



UNIVERSIDAD DEL ROSARIO

# AN ACCURATE HESTON IMPLEMENTATION WITH USD-COP DATA

Javier Jaher Alfonso Lázaro Salcedo

Universidad del Rosario  
Facultad Economía  
Bogotá, Colombia  
2018

—

T

# AN ACCURATE HESTON IMPLEMENTATION WITH USD-COP DATA

JAVIER JAHER ALFONSO LÁZARO SALCEDO  
ADVISOR: RAFAEL SERRANO

*These words are dedicated specially to the memory of my dad, and to my mom,  
who never stop believing in me.*

**ABSTRACT.** This study find by empirical evidence a fast and accurate way to calculate the price of a European Call using the Heston (1993) model.

It calculate and uses a benchmark price calculated with the mentioned Heston 1993 pricing approaches and the trapezoidal rule with  $a = 1e-20000$ ;  $b = 300$ ;  $N = 10000000$ , to find which combination of Heston pricing process and numerical schemes leads to a computationally faster and more accurate price process. Two equivalent pricing methods and seven numerical schemes are calculated in order to find wich combination take less time to be compute and is closes to the benchmark as posible. The study uses  $Q$ -measure in the sense of spot data, and the other  $P$ -measure in the sense of historical data. That mean the study calculate two parameter sets. one under mesure  $Q$  and other under  $P$  by Maximum Likelihood and non-linear least square function, respectively, to somehow proof the conclusion dose not depends on how the parameter are found. Study stands that the accuraste way to calculate the Heston price in the Colombian FX market data used is consolidating the integrals for the probability  $P1$  and  $P2$  that the original framework propose and solve the integral using Gauss-Legendre or Gauss-Laguerre.

---

*Date:* February 15, 2018.

*2010 Mathematics Subject Classification.* 65D30.

*Key words and phrases.* Heston model, USD-COP, Fourier pricing, Fourier transformation, Fundamental transform, Damping factor, Quantitative methods, Gaussian quadrature, Newton cotes.

Thanks to Dana. In the distance you gave me strength. Special thanks to my cousin Dasaed for the patience that he has always had with my English. Hope this sentence is grammatically okay Dasa.

## 1. INTRODUCTION

In developed markets, the Heston model and a wide variety of more complex models have been implemented to make better and accurate valuations of options so that affordable Hedging Strategies can be made. The drawback of these models is that the advantage of a more realistic modeling of the variance can be offset by the costs of calibration and implementation due to the fact that closed-form formulas are rarely available like in *BSM* (Black-Scholes-Merton) model. See [GT11].

In an emerging market such as the one in Colombia, those more complex models are implemented by a few as a result of the market liquidity. In other words, if the profit of use de profit of selling options at the Heston price in a hedge strategy is not attractive to investors, the market hardly will improve the liquidity. Investors should have incentives to move their money from colombian equities and fixed income to the derivatives market. The reader may find usefull the process of finding the price of an european call option in the Colombian FX market computationally faster and accurate so a better replicating portfolio can be used.

In order to achive the goal, the study shows how to implement the Heston model, a more complex model than *BSM*, with two equivalent approaches and seven different numerical schemes to find wich combination leads to a more computationally faster and accurate valuation of call options in the Colombian FX market. Think about to equivalent "formulas" that have one or two integrals, and in order to solve them, the study present seven diferent ways to do it.

The to equivalent pricing approaches or "formulas" are (i) the Hestons original paper and its Characteristic function, (ii) Consolidating the integrals for the probability P1 and P2 that the original framework propose. In the Apendix, the interest reader can found other evivalent pricing method

Diferent ways to solve the integral are presented and each of them are called numerical scheme. The first four are Newton Cotes formulas (i)Mid-Point (ii) Trapezoidal rule, (iii) Simpson's rule, (iv) Simpson's 3/8 rule. The remaining schemes are Gaussian Cuadratures: (v) Gauss-Laguerre, (vi) Gauss-Legendre, (vii) Gauss-Lobato.

Furthermore, this study calculate de price under the mesure  $Q$  and  $P$  to show in some way that the conclusion does not depend on the measure used. In order to do that, the paremeter set  $\Theta = \{\kappa, \theta, \sigma, v_0, \rho\}$  is estimated by Maximum Likelihood (hereafter *MLE*)[AW09] under the risk neutral mesure  $Q$  and also by calibration using a non-linear least square function (hearafter *NLLS*) under mesure  $P$ . The study will go into this topic later.

Evenmore, in one hand, to find the accurate solution, the study calculate a benchmark price that have at least 41 decimal<sup>1</sup> and compare all prices with those in order to find which combination of price approach and numerical scheme have the smallest error. In the other, to find the computationally faster price approach and numerical scheme the time the code took to calculate de price is averaged and compere. How the benchmark is calculated and chosen will be explained in chapter 9.

The data used in this study was chosen in a trading session without high impact of political or macronomic news that could afected the price of the underlying. Likewise, the correlated currencies did not have sudden movements compared with the dollar. The data used in this study was from July 12, 2016. This day opened at 2,935 COP/USD and closed at 2,918 COP/USD.

12/07/2016 COP/USD Market Data							
Market IV						RF	RD
	10D Put USD	25D Put USD	ATM	25D Call USD	10D Call USD		
<b>1W</b>	19,115	16,393	16,405	17,778	21,335	0,294%	8,797%
<b>1M</b>	16,561	16,069	16,71	18,051	19,919	0,357%	7,181%
<b>2M</b>	15,479	15,138	15,883	17,358	19,326	0,373%	7,604%
<b>3M</b>	15,084	14,629	15,417	17,166	19,466	0,377%	7,975%
<b>6M</b>	14,856	14,15	14,937	16,9	19,644	0,413%	7,493%
<b>9M</b>	14,232	13,452	14,29	16,417	19,427	0,433%	7,229%
<b>1Y</b>	13,795	12,984	13,85	16,156	19,46	0,452%	7,370%

FIGURE 1. Market Data Used

Finally, this study uses Rouag book [RH13] as a detailed guide. The mathematical formulas for the pricing calculation as well as a big part of the code were heavily supported on his book. Thanks to his awesome

<sup>1</sup>To fulfill the study goal, it will be assumed that the level of accuracy of the benchmark will absolute because it is going to be the initial assumption to find the most accurate price

work this study is possiblee

**1.1. How is the Study Organized.** The study is composed of eighth chapters. The first one is the already mentioned introduction. The second one focuses on the theoretical framework and the approaches. Chapter three is oriented to the numerical schemes that solved the remaining integral. The FX Market Conventions are explain in chapter four In chapter five and six the estimation and calibration topic is covered respectively. Chapter seven checks how the benchmark is defined. Chapter eighth proofs wich pricing approach has the fastest numerical scheme with the most accuracy. Chapter nine ilustrated how the study find the most accurate method. Chapter ten illustrate how the accurate method and faster scheme are used to reach the conclusion of the study. Finally, chapter eighth presents the conclusions.

## 2. THE HESTON MODEL

This chapter presents the original Heston framework and introduces the reader to other six equivalent pricing approaches, presented in the order they where published.

**2.1. Heston 1993.** The Heston (1993) models the underling with two SDE, one for the price,  $S_t$  and one for the variance  $v_t$ .

$$(2.1.1) \quad \begin{aligned} dS_t &= \mu S_t dt + \sqrt{v_t} S_t dW_{1,t} \\ dv_t &= \kappa(\theta - v_t)dt + \sigma \sqrt{v_t} dW_{2,t} \\ E^P[dW_{1,t} dW_{2,t}] &= \rho dt \end{aligned}$$

Where:

$$(2.1.2) \quad \begin{array}{ll} \mu & \text{drift process of the underling,} \\ \kappa > 0 & \text{mean reversion speed of the variance,} \\ \theta > 0 & \text{mean reversion level of thevariance,} \\ \sigma > 0 & \text{volatility of volatility,} \\ v_0 > 0 & \text{initial level of volatility,} \\ \lambda & \text{volatility risk parameter.} \end{array}$$

Hence, the model is constituted by a bivariate system of SDE where  $W_{1,t}$  has a correlation  $\rho \in [-1 \ 1]$  with  $W_{2,t}$  and is expected to be positive in the USD-COP FX market since investors seek shelter in the

strongest foreign exchange when  $v_t$  increases. The correlation parameter,  $\rho$ , controls the skewness of the density of the logarithm of the underlying, when its positive the probability density will be positive skewed.

Under the risk neutral measure  $\mathbb{Q}$  the model stand:

$$(2.1.3) \quad \begin{aligned} dS_t &= rS_t dt + \sqrt{v_t} S_t d\tilde{W}_{1,t} \\ dv_t &= [\kappa(\theta - v_t) - \lambda_{S_t, v_t, t}] dt + \sigma \sqrt{v_t} d\tilde{W}_{2,t} \end{aligned}$$

where  $\tilde{W}$  is the Brownian motions under the risk-neutral process,  $\lambda_{S_t, v_t, t}$  is the the volatility risk parameter.  $\lambda^2$ , is set to zero because it is embedded into  $\kappa^*$  and  $\theta^*$ :

$$(2.1.4) \quad \begin{aligned} \lambda_{S_t, v_t, t} &= \lambda v \\ dS_t &= rS_t dt + \sqrt{v_t} S_t d\tilde{W}_{1,t} \\ dv_t &= \kappa^*(\theta^* - v_t) dt + \sigma \sqrt{v_t} d\tilde{W}_{2,t} \\ E^{\mathbb{Q}}[d\tilde{W}_{1,t} d\tilde{W}_{2,t}] &= \rho dt \end{aligned}$$

where  $\kappa^* = \kappa + \lambda$  and  $\theta^* = \frac{\kappa\theta}{\kappa + \lambda}$

When estimating the risk-neutral parameters. For notation simplicity the asterisk will be drop and it will be understood hereafter that the study is dealing with the risk-neutral measure.

The characteristic function (hereafter CF) approach to option pricing of Heston 1993 can be applied to the characterization of call prices in the form of discounting the expected value of the payoff function under the risk-neutral measure as:

$$(2.1.5) \quad \begin{aligned} C(k) &= e^{-rd\tau} E^{\mathbb{Q}}[(S_t - K)^+] \\ &= e^{-rd\tau} E^{\mathbb{Q}}[(S_t - K) \mathbf{1}_{S_t > K}] \\ &= e^{-rd\tau} E^{\mathbb{Q}}[S_t \mathbf{1}_{S_t > K}] - K e^{-rd\tau} E^{\mathbb{Q}}[\mathbf{1}_{S_t > K}] \\ &= S_t e^{-rf\tau} P_1 - K e^{-rd\tau} P_2 \end{aligned}$$

where the quantities  $P_1$  and  $P_2$  represent the probability of the option expiring ITM conditional on the filtration,  $\mathcal{F}_t$ , under the measure  $\mathbb{Q}$ . In other words  $P_1$  uses the underlying as numerarie while  $P_2$  uses

---

<sup>2</sup> Estimation of  $\lambda$  is subject-matter to its own research. See Bollerslev et al. (2011)[BGZ11]

the bond,  $B_t$ . Bakshi and Madan (2000)[BM00] prove that the derivation of the call price under change of numeraire is valid for the *BSM* and Heston model.

In the *BSM* world,  $P1 = \phi(d1)$  and  $P2 = \phi(d2)$  are calculated straightforward, on the other hand, for the Heston model to obtain such probabilities the inversion theorem of the CF of Gil-Pelaez (1951) [GP51] must take place. For details on pricing with CF see Zhu (2009) [Zhu09]. To implement the inversion theorem, the CF must be known. Heston (1993) proposes the following CF:

$$(2.1.6) \quad f_j(\phi; x_t, v_t) = \exp( C_j(\tau, \phi) + D_j(\tau, \phi)v_t + i\phi x_t )$$

for  $j = 1, 2$ . Heston (1993) stands that the CF for the log-returns,  $x_T = \ln S_T$ , is a way to exploit the linearity coefficient on the model PDE (Partial Differential Equation). The calculation of the Heston PDE is slightly more difficult than *BSM* PDE. The interested reader can follow up the derivation in Rouah (2013) to get:

$$(2.1.7) \quad \begin{aligned} \frac{\partial P_j}{\partial j} + \rho\sigma v \frac{\partial^2 P_j}{\partial v \partial x} + \frac{1}{2}v \frac{\partial^2 P_j}{\partial x^2} + \frac{1}{2}\sigma^2 v \frac{\partial^2 P_j}{\partial v^2} \\ + (r + \mu_j v) \frac{\partial P_j}{\partial x} + (a - b_j v) \frac{\partial P_j}{\partial v} = 0. \end{aligned}$$

where  $\mu_1 = \frac{1}{2}, \mu_2 = -\frac{1}{2}, a = \kappa\theta, b_1 = \kappa + \lambda - \rho\sigma, b_2 = \kappa + \lambda$ . The CF will follow the PDE (2.1.7) as a consequence of the Feynman-Kac theorem that stipulates the solution  $f(\phi; x_t, v_t) = E[e^{i\phi \ln S_T} | x_t, v_t]$ , which is the CF for  $X_T = \ln S_T$ . Once it is known that the PDE (2.1.7) can be applied to the CF, one may proceed.

To find the coefficients of the CF (2.1.6) one must express the PDE (2.1.7) for the CF<sup>3</sup>, express the six partial derivate of the new PDE in terms of the solution proposed by Heston (1993) (2.1.6), replace in the new PDE, solve the remaining Riccati equation  $D_j$  and the ordinary differential equation  $C_j$  in order to get:

$$(2.1.8) \quad \begin{aligned} D_j(\tau, \phi) &= \frac{b_j - \rho\sigma i\phi + d_j}{\sigma^2} \left( \frac{1 - e^{d_j\tau}}{1 - g_j e^{d_j\tau}} \right), \\ C_j(\tau, \phi) &= ri\phi\tau + \frac{a}{b} \left[ (b_j - \rho\sigma i\phi + d_j)\tau - 2\ln \left( \frac{1 - g_j e^{d_j\tau}}{1 - g_j} \right) \right], \end{aligned}$$

---

<sup>3</sup>change  $P_j$  for  $f_j$

where  $g_j = \frac{b_j - \rho\sigma i\phi + d_j}{b_j - \rho\sigma i\phi - d_j}$ ,  $d_j = \sqrt{(\rho\sigma i\phi - b_j)^2 - \sigma^2(2\mu_j i\phi - \phi^2)}$  for  $j=1,2$ .

Note that the CF does not depend on the strike, but it does on the maturity,  $\tau$ . This implies, when computing the Non-linear least square function for calibration purposes, the values for  $f_1(\phi)$  and  $f_2(\phi)$  can be calculated only once for each maturity and must be used repeatedly across the deltas.<sup>4</sup> This method will save computational time, because by far, the CF is the most time consuming operation. In spite of the advantages of the mentioned method the loss of accuracy in the parameters can affect the results of the study and will not be implemented. For details on accelerating the calibration see Kilin (2006) [Kil06].

The disadvantage of the CF proposed by Heston (1993) is the discontinuities at some points. Albrecher et al. (2007) [AMS] propose a CF in (2.1.9) that is equivalent to the Heston (1993) (2.1.8) but causes less numerical problems. For the derivation of the "Heston little trap" CF, one must multiply  $D_j$  by  $\exp(-d_j\tau)$  in the numerator and denominator and take out from the logarithm in  $C_j$  the term  $\exp(d_j\tau)$ , and make some algebraic operations to express the logarithm in terms of  $c_j$ :

$$(2.1.9)$$

$$D_j(\tau, \phi) = \frac{b_j - \rho\sigma i\phi - d_j}{\sigma^2} \left( \frac{1 - e^{-d_j\tau}}{1 - c_j e^{-d_j\tau}} \right),$$

$$C_j(\tau, \phi) = r i\phi\tau + \frac{a}{b} \left[ (b_j - \rho\sigma i\phi - d_j)\tau - 2\ln \left( \frac{1 - c_j e^{-d_j\tau}}{1 - c_j} \right) \right].$$

where  $c_j = \frac{1}{g_j} = \frac{b_j - \rho\sigma i\phi - d_j}{b_j - \rho\sigma i\phi + d_j}$ .

In this accurate Heston implementation, the CF proposed by Albrecher et al. (2007) is always used. Once the CF is defined, applying the inversion theorem of Gil-Pelaez (1951) the probabilities  $P1$  and  $P2$  are obtained:

$$(2.1.10)$$

$$P_j = Pr(\ln S_T > \ln K) = \frac{1}{2} + \frac{1}{\pi} \int_0^\infty \left[ \frac{e^{-i\phi \ln K} f_j(\phi; x, v)}{i\phi} \right] d\phi$$

It is worth mention that the integral will be compute in the real part.

---

<sup>4</sup>In FX market the quoted options are in terms of deltas not strikes. Later chapters focus on this.



Therefore the Heston (1993) price equation needs to compute the probabilities  $P_j$  for  $j=1,2$ . and apply them on (2.1.5).

This study uses the following sniped code 2 to calculate de Heston (1993) price.

```
function [Price,Domain,Npoints] = Price93(PutCall,S,K,T,rf,rd,kappa,theta,sigma,lambda,v0,rho,I

% Heston (1993) Call price Newton-Coates Formulas, Gauss-Laguerre, ...

Domain = [a b];
Npoints = N;
% Define P1 and P2
P1 = 1/2 + 1/pi*sum(int1);
P2 = 1/2 + 1/pi*sum(int2);

%Heston Call
HestonC = S*exp(-rd*T)*P1 - K*exp(-rf*T)*P2;

% Output the option price
if strcmp(PutCall,'C')
    Price = HestonC;
else
    %Heston Put by put-Call-Parity
    Price = HestonC - S*exp(-rd*T) + K*exp(-rf*T);
```

FIGURE 2. Heston(1993) Code

**2.2. Consolidating the integrals of Heston (1993).** Since the integrals are virtually the same, equal domain,  $[0 \infty]$  and integration variable,  $\phi$ , it is possible to express the Heston price under one integral. In other words, the probabilities  $P1$  and  $P2$  can be joint up into a single integral which speed up the numerical integration.

(2.2.1)

$$C(K) = \frac{1}{2}S_te^{-rf\tau} - \frac{1}{2}Ke^{-rd\tau} + \frac{1}{\pi} \int_0^\infty Re \left[ \frac{e^{-i\phi \ln K}}{i\phi} (S_te^{-rf\tau} f_1(\phi; x, v) - Ke^{-rd\tau} f_2(\phi; x, v)) \right] d\phi$$

The advantage of this pricing method is reduced computational time by almost one-half. For more details see [RH13].

This study uses the following sniped code 3 to calculate the integral

And then uses 4 to calculate the Consolidating the integrals of Heston (1993) price

Now that all the Heston prices approaches have been presented, the numerical schemes must be introduced. The numerical schemes will be treated in the following section.

```

function y = Consoli_Integrand(phi,kappa,theta,lambda,rho,sigma,tau,K,S,rd,rf,v0)
% Returns the integrand for the risk neutral probabilities P1 and P2

% Log of the stock price
x = log(S);

% Parameter "a" is the same for P1 and P2.
a = kappa*theta;

% First characteristic function f1
u1 = 0.5;
b1 = kappa + lambda - rho*sigma;
d1 = sqrt((rho*sigma*i*phi - b1)^2 - sigma^2*(2*u1*i*phi - phi^2));
g1 = (b1 - rho*sigma*i*phi + d1) / (b1 - rho*sigma*i*phi - d1);

% "Little Heston Trap" formulation
c1 = 1/g1;
D1 = (b1 - rho*sigma*i*phi - d1)/sigma^2*((1-exp(-d1*tau))/(1-c1*exp(-d1*tau)));
G1 = (1 - c1*exp(-d1*tau))/(1-c1);
C1 = (rd-rf)*i*phi*tau + a/sigma^2*((b1 - rho*sigma*i*phi - d1)*tau - 2*log(G1));

f1 = exp(C1 + D1*v0 + i*phi*x);

% Second characteristic function f2
u2 = -0.5;
b2 = kappa + lambda;
d2 = sqrt((rho*sigma*i*phi - b2)^2 - sigma^2*(2*u2*i*phi - phi^2));
g2 = (b2 - rho*sigma*i*phi + d2) / (b2 - rho*sigma*i*phi - d2);

% "Little Heston Trap" formulation
c2 = 1/g2;
D2 = (b2 - rho*sigma*i*phi - d2)/sigma^2*((1-exp(-d2*tau))/(1-c2*exp(-d2*tau)));
G2 = (1 - c2*exp(-d2*tau))/(1-c2);
C2 = (rd-rf)*i*phi*tau + a/sigma^2*((b2 - rho*sigma*i*phi - d2)*tau - 2*log(G2));

f2 = exp(C2 + D2*v0 + i*phi*x);

% Return the real part of the integrand.
y = real(exp(-i*phi*log(K))/i/phi*(S*exp(-rf*tau)*f1 - K*exp(-rd*tau)*f2));

```

FIGURE 3. Consolidating the integrals of Heston (1993) Code

### 3. NUMERICAL SCHEMES

Due to the non existence of the anti-derivate of the Heston integrand, the probabilities  $P_{1,2}$  must be approximated numerically. This task is challenging. The first challenge lies in the CF not being defined at zero, even though the integration domain is  $[0, \infty)$ , meaning that the lower boundary of the integral is represented by a very small number and the upper boundary for a number that represents infinity. In others words the domain  $[0, \infty)$  will be represented by  $[a, b]$  where  $a$  should be close to zero, and  $b$  is as big as desired.

```

function [Price,Domain,Npoints] = PriceConsoli(PutCall,S,K,T,rd,rf,kappa,theta,sigma,lambda,v0,rho,IntRule,a,b,N)
% Regroup the integrals for the probabilities P1 and P2 into a single
Domain = [a b];
Npoints = N;

% Integration methods
if IntRule==1
    % Mid-Point rule -----
    h = (b-a)/(N-1);
    phi = a:h:b;
    wt = h.*ones(1,N);
    for k=1:N-1;
        mid(k) = (phi(k) + phi(k+1))/2;
        int(k) = wt(k)*Consoli_Integrand(mid(k),kappa,theta,lambda,rho,sigma,T,K,S,rd,rf,v0);
    end
elseif IntRule==7
    % Gauss Lobatto [-1 1]-----
    for k=1:length(xGLo);
        X = (a+b)/2 + (b-a)/2*xGLo(k);
        int(k) = wGLo(k)*Consoli_Integrand(X,kappa,theta,lambda,rho,sigma,T,K,S,rd,rf,v0)*(b-a)/2;
    end
    Npoints = length(xGLo);
end

%Heston Consolidating integral
HestonC = (1/2)*S*exp(-rf*T) - (1/2)*K*exp(-rd*T) + sum(int)/pi;

% Output the option price
if strcmp(PutCall,'C')
    Price = HestonC;
else
    % The put price by put-call parity
    Price = HestonC - S*exp(-rf*T) + K*exp(-rd*T);
end

```

FIGURE 4. Calculating the price of Consolidating the integrals of Heston (1993) Code

The second challenge lies on the oscillation of the integrand. For some parameters, maturities, and deltas<sup>5</sup>, the integrand can oscillate wildly, implying that it does not decay quickly to zero and a large number for the upper boundary will be needed. In general, the oscillation and speed of decay vary inversely with maturity. For example, in a one week option, the integrand will oscillate much more than the one year option. See figure 22 vs 24 to see the oscillation. The reason behind the oscillation in short maturities is the limitation of not only the Heston model, but the stochastic volatility models as well. However, this can be solved by adding jumps to the pricing process, see Bates (1996)

<sup>5</sup>Strikes are expressed in terms of delta in the FX market

[Bat96].

In order to solve the previous challenge, the right choice of  $b$  must be made.  $b$  can be chosen with the MD (multi-domain) integration approach of Zhu (2010) to select the upper limit. In this integration approach the domain  $[0, \infty)$  will be represented by  $[a, b]$ , where  $a$  should be close to zero, and  $b$  is as big as desired. Once the integration interval where the numerical method will be applied is defined, it is split into  $N$  parts and each part is integrated as a whole. The probability is calculated as the sum of the area under the curve of each subinterval, but only when the subinterval area is greater than a determined tolerance level. Otherwise, the subinterval area is not taken into account and the calculation will stop.

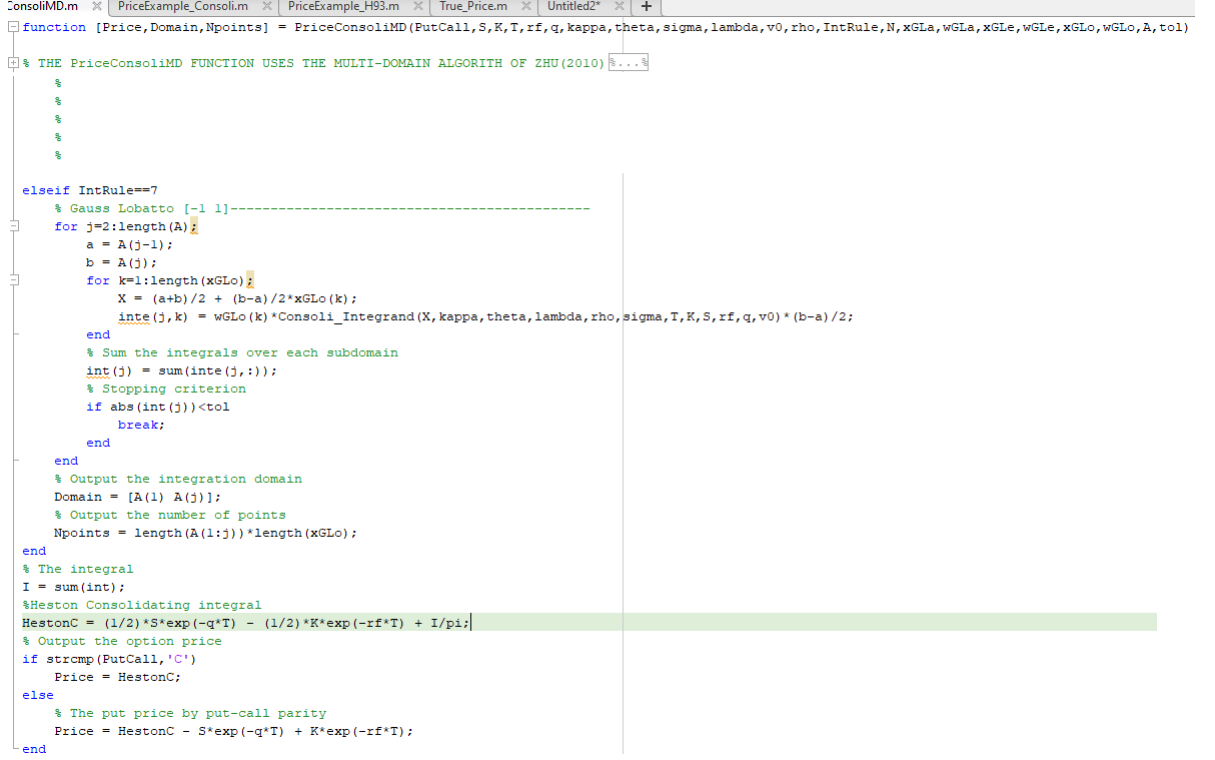
As an example, one can consider the following case: an interval  $[a = 1e - 5, b = 150]$  used to represent  $[0, \infty)$ ,  $N = 3$  and a tolerance level of  $1e - 6$ <sup>6</sup>. To implement the MD method, the interval is first split into  $N$  parts,  $[1e - 5, 50]$ ,  $(50, 100]$ ,  $(100, 150]$ . The first interval is integrated by the decided method, and it yields a result greater than the tolerance level, meaning it will be summed up. The same procedure applies to the next intervals, and in this case, the result of the second interval is also greater than the tolerance. However, when the last interval is integrated, the resulting value is less than the tolerance, and therefore it contributes little to the area. In other words, the integral decays to zero near 100, and therefore the last interval will not be taken into account. At this point the MD method will stop. In the example, the integration domain was reduced from  $[1e - 5, 150]$  to  $[1e - 5, 100]$  due to the MD. See figure5.

This method will assign a wider domain for shorter maturities and a narrower for longer ones. Zhu (2010) states that this is an optimal method for assigning the upper limit of the integral. However, if  $N$  increases, the computational time increases as well, due to the fact that instead of making one integral, it now has to make at least  $N$  integrals. Furthermore, the *ad hoc* choice of the upper limit must always be done since the interval must be defined in all cases. The most accurate way, regardless of the time, is to plot the integrates  $P_{1,2}$  for all the maturities and all the deltas, choose  $b$  based on the graph, and apply the MD method with a large  $N$  value. It is worth noting

---

<sup>6</sup>Using a small tolerance level of  $\leq 1e - 6$  with  $M > 1e6$  subintervals of area could lead to an error of order 1

that other methods for finding  $b$  exist, such as the one in lewis (2000):  
 $b = \max[1000, 10/\sqrt{v_0\tau}]$ .



```

ConsoliMD.m x PriceExample_Consoli.m x PriceExample_H93.m x True_Price.m x Untitled2* x +
function [Price, Domain, Npoints] = PriceConsoliMD(PutCall, S, K, T, rf, q, kappa, theta, sigma, lambda, v0, rho, IntRule, N, xGLa, wGLa, xGLE, wGLE, xGLO, wGLO, A, tol)

% THE PriceConsoliMD FUNCTION USES THE MULTI-DOMAIN ALGORITHM OF ZHU(2010)

%
%
%
%

elseif IntRule==7
    % Gauss Lobatto [-1 1]-----
    for j=2:length(A)
        a = A(j-1);
        b = A(j);
        for k=1:length(xGLO)
            X = (a+b)/2 + (b-a)/2*xGLO(k);
            inte(j,k) = wGLO(k)*Consoli_Integrand(X, kappa, theta, lambda, rho, sigma, T, K, S, rf, q, v0) * (b-a)/2;
        end
        % Sum the integrals over each subdomain
        int(j) = sum(inte(j,:));
        % Stopping criterion
        if abs(int(j))<tol
            break;
        end
    end
    % Output the integration domain
    Domain = [A(1) A(j)];
    % Output the number of points
    Npoints = length(A(1:j))*length(xGLO);
end
% The integral
I = sum(int);
%Heston Consolidating integral
HestonC = (1/2)*S*exp(-q*T) - (1/2)*K*exp(-rf*T) + I/pi;
% Output the option price
if strcmp(PutCall, 'C')
    Price = HestonC;
else
    % The put price by put-call parity
    Price = HestonC - S*exp(-q*T) + K*exp(-rf*T);
end

```

FIGURE 5. Multi Domain Function

The third challenge lies in the discontinuity of the CF over the domain  $[0, \infty)$ . To solve this problem, the CF of Albrecher et al. (2007) was applied. Since "The little Heston Trap" always works, other more complex alternatives to overcome this problem, like the rotation algorithm of Kahl and Jäckel (2005) [KJ05] and the ones proposed on Zhu (2010) are not taken into account. See figure 6.

The numerical methods taken into account were Newton-Cotes (hereafter NC) rules and Gaussians Quadratures (hereafter GQ). Both of these approximate the remaining Heston integral over the domain  $[a, b]$  by equation (3.0.1). They calculate the area as the sum of the function evaluated at the abscissas,  $x_1, \dots, x_N$ , multiplied by their respective weight  $w_1, \dots, w_N$ .

```

function y = Consoli_Integrand(phi, kappa, theta, lambda, rho, sigma, tau, K, S, rd, rf, v0)
% Returns the integrand for the risk neutral probabilities P1 and P2 [...]
% Log of the stock price
x = log(S);

% Parameter "a" is the same for P1 and P2.
a = kappa*theta;

% First characteristic function f1
u1 = 0.5;
b1 = kappa + lambda - rho*sigma;
d1 = sqrt((rho*sigma*phi - b1)^2 - sigma^2*(2*u1*phi - phi^2));
g1 = (b1 - rho*sigma*phi + d1) / (b1 - rho*sigma*phi - d1);

% "Little Heston Trap" formulation
c1 = 1/g1;
D1 = (b1 - rho*sigma*phi - d1)/sigma^2*((1-exp(-d1*tau))/(1-c1*exp(-d1*tau)));
G1 = (1 - c1*exp(-d1*tau))/(1-c1);
C1 = (rd-rf)*phi*tau + a/sigma^2*((b1 - rho*sigma*phi - d1)*tau - 2*log(G1));

f1 = exp(C1 + D1*v0 + phi*x);

% Second characteristic function f2
u2 = -0.5;
b2 = kappa + lambda;
d2 = sqrt((rho*sigma*phi - b2)^2 - sigma^2*(2*u2*phi - phi^2));
g2 = (b2 - rho*sigma*phi + d2) / (b2 - rho*sigma*phi - d2);

% "Little Heston Trap" formulation
c2 = 1/g2;
D2 = (b2 - rho*sigma*phi - d2)/sigma^2*((1-exp(-d2*tau))/(1-c2*exp(-d2*tau)));
G2 = (1 - c2*exp(-d2*tau))/(1-c2);
C2 = (rd-rf)*phi*tau + a/sigma^2*((b2 - rho*sigma*phi - d2)*tau - 2*log(G2));

f2 = exp(C2 + D2*v0 + phi*x);

% Return the real part of the integrand.
y = real(exp(-phi*log(K))/phi*(S*exp(-rf*tau)*f1 - K*exp(-rd*tau)*f2));

```

FIGURE 6. Multi Domain Function

$$(3.0.1) \quad \int_a^b f(x) dx \approx \sum_{j=1}^N w_j f(x_j)$$

The NC rules are easy to understand since they are the simplest integration rules. Unfortunately, they require the most computational time. Using equation (3.0.1) to defined the NC rules implies that the abscissas are equidistant, making the integration interval equality spaced. This implies that computational time dramatically increases for an accurate integration because many abscissas are needed. The drawback of NC rules lies in calculating the weights since they are not equal for all the abscissas and are method depending. The weights are also dependent of N, the numbers of parts the intervals must be split in.

On the other hand, the GQ abscissas and weights, beside from being less, are unequally spaced and calculated by functions depending on the method. Each quadrature has one equation for the abscissas and one for the weight. The abscissas are specified in advance, so the upper and lower boundary problem is solved. The disadvantage with this

method lies on its complexity, however, once the abscissas and weights are calculated, applying the method becomes a straightforward and trivial exercise. The GQ substantially reduces the computational time vs NC due to the facts that the numerical method adapts to some how fit the integral. The value of the abscissas and weights depend on the choice of  $N$ . In this study,  $N$  will be 32 for all GQ, following the guidance of Rouah (2014).

To numerically evaluate the Heston integrals  $P_{1,2}$  using the proposed numerical methods, the abscissas and weights must be calculated first. Then, as the following step, the integrand is evaluated at each abscissa and the result is multiplied by the corresponding weight. Once this is done, the result is store in a vector and all the terms of the vector must be added in order to calculate the integral. In other words, apply equation (3.0.1).

The numerical methods that were taken into account were: four NC rules and three GQ. The following subsections introduce all the mentioned numerical methods. The literature on numerical methods is rich and there are excellent text books like Burden and Faires (2010) [BF10] and Abramowitz and Stegun (1964), [AS64] which the reader may check for further details.

**3.1. Mid-Point.** It approximates the integral as the sum of rectangles, each with the same width,  $x_{j+1} - x_j$ , and height equal to the integrand evaluated at the mid point of the interval width. The Mid-point rule is defined as:

$$(3.1.1) \quad \int_a^b f(x) dx \approx h \sum_{j=1}^{N-1} f\left(\frac{x_j + x_{j+1}}{2}\right)$$

where the abscissas are defined as  $x_j = a + (j - 1)h$  and the weights as  $w_j = h = (b - a)/(N - 1)$ , with  $x_1 = a$ ,  $x_N = b$ .

The code used in this study to implement the Mid-Point numerical scheme 7 to calculate de Heston (1993) price is:

**3.2. Trapezoidal Rule.** It approximate the integral as the sum of trapezoids, each with equal width,  $x_{j+1} - x_j$ , and the height equal to the integrand evaluated at the end point of each subinterval. The trapezoids are constructed by drawing a segment between  $f(x_j)$  and  $f(x_{j+1})$ . The formula for the Trapezoidal rule is:

```

% Integration methods
if IntRule==1
    % Mid-Point rule -----
    h = (b-a)/(N-1);
    phi = a:h:b;
    wt = h.*ones(1,N);
    mid = zeros(1,N-1);
    int1 = zeros(1,N-1);
    int2 = zeros(1,N-1);
    for k=1:N-1;
        mid(k) = (phi(k) + phi(k+1))/2;
        int1(k) = wt(k) * HestonProb(mid(k), kappa, theta, lambda, rho, sigma, T, K, S, rf, rd, v0, 1);
        int2(k) = wt(k) * HestonProb(mid(k), kappa, theta, lambda, rho, sigma, T, K, S, rf, rd, v0, 2);
    end
end

```

FIGURE 7. Mid-Point Code

$$(3.2.1) \quad \int_a^b f(x) dx \approx \frac{h}{2}f(x_1) + h \sum_{j=1}^{N-1} f(x_j) + \frac{h}{2}f(x_N)$$

where the abscissas and  $h$  are defined as the Mid-point rule.  $w_1 = w_N = h/2$  and  $w_j = h$  for  $j = 2, \dots, N-1$ .

The code used in this study to implement the Mid-Point numerical scheme 8 to calculate de Heston (1993) price is:

```

elseif IntRule==2
    % Trapezoidal rule -----
    h = (b-a)/(N-1);
    phi = a:h:b;
    wt = h.*[1/2 ones(1,N-2) 1/2];
    for k=1:N;
        int1(k) = wt(k) * HestonProb(phi(k), kappa, theta, lambda, rho, sigma, T, K, S, rf, rd, v0, 1);
        int2(k) = wt(k) * HestonProb(phi(k), kappa, theta, lambda, rho, sigma, T, K, S, rf, rd, v0, 2);
    end
end

```

FIGURE 8. Trapezoidal Rule Code

**3.3. Simpson's Rule.** Each NC rule is more complicated than the previous one and less than the subsequent. This NC uses quadratic polynomials in the approximation, following Rouah (2014). The integral is defined as:

$$(3.3.1) \quad \int_a^b f(x) dx \approx \frac{h}{3}f(x_1) + \frac{4h}{3} \sum_{j=1}^{\frac{N}{2}-1} f(x_{2j}) + \frac{2h}{3} \sum_{j=1}^{\frac{N}{2}} f(x_{2j-1}) + \frac{h}{3}f(x_N)$$



where the abscissas and  $h$  are defined as the Mid-point and Trapezoidal rule, but with  $w_1 = w_N = h/3$  along with  $w_j = 4h/3$  when  $j$  is even, and  $w_j = 2h/3$  when is odd.

The code used in this study to implement the Mid-Point numerical scheme 9 to calculate de Heston (1993) price is:

```
elseif IntRule==3
% Simpson's Rule -----
h = (b-a)/(N-1);
phi = a:h:b;
wt = (h/3).*[1 (3+(-1).^(2:N-1)) 1];
for k=1:N;
    int1(k) = wt(k)* HestonProb(phi(k),kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,1);
    int2(k) = wt(k)* HestonProb(phi(k),kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,2);
end
```

FIGURE 9. Simpson's Rule Code

**3.4. Simpson's 3/8 Rule.** This rule is a refinement of the previous one. It uses cubic polynomials in the approximation of the integral and is the more sophisticated NC rule in this study. It is defined as:

(3.4.1)

$$\int_a^b f(x) dx \approx \frac{3h}{8}f(x_0) + \frac{6h}{8} \sum_{j=3,6,9,\dots}^{N-3} f(x_j) + \frac{9h}{8} \sum_{j \neq 3,6,9,\dots}^{N-1} f(x_j) + \frac{3h}{8}f(x_N)$$

Note that this rule starts with  $x_0$  because the abscissas are defined as:  $x_j = a + ih$  for  $i = 0, \dots, N$  where  $N$  is divisible by three and with  $h = (b - a)/N$ . The weights depend on whether  $j$  is divisible or not by three:

$$w_j = \begin{cases} 3h/8 & \text{if } j = 0 \text{ or } j = N \\ 6h/8 & \text{if } j = 3, 6, 9, \dots \\ 9h/8 & \text{if } j \neq 3, 6, 9, \dots \end{cases}$$

The implementation of all the NC rules presented here are straightforward, although the calculation of the weights may be tricky at times. For the Simpson's Rule and Simpson's 3/8 rule one must be careful with the fact that  $N$  must be divisible by three and that this will directly affect calculation of the weights.

The code used in this study to implement the Simpson's Rule 3/8 numerical scheme 10 to calculate de Heston (1993) price is:

```

elseif IntRule==4
    % Simpson's 3/8 rule -----
    % Ensure that N-1 is divisible by 3
    N = N-mod(N,3)+1;
    h = (b-a)/(N-1);
    phi = a:h:b;
    wt = (3*h/8).*[[1 3 3] repmat([2 3 3],1,(N-1)/3-1) 1];
    for k=1:N
        int1(k) = wt(k) * HestonProb(phi(k),kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,1);
        int2(k) = wt(k) * HestonProb(phi(k),kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,2);
    end

```

FIGURE 10. Simpson's Rule 3/8 Code

**3.5. Gauss-Laguerre.** This quadrature is specially relevant because it is designed for the domain  $[0, \infty)$ . Its abscissas are the roots of the Laguerre polynomial,  $L_N(x)$ , defined as:

$$(3.5.1) \quad L_N(x) = \sum_{k=0}^N \frac{(-1)^k}{k!} \binom{N}{k} x^k$$

where the last term in (3.5.1) is the binomial coefficient. The weights function uses the derivative of  $L_N(x)$  evaluated at each abscissas in the following equation:

$$(3.5.2) \quad w_j = \frac{(n!)^2 e^{x_j}}{x_j [L'_N(x_j)]^2}$$

The code used in this study to implement the Gauss-Laguerre numerical scheme 11 to calculate de Heston (1993) price is:

```

elseif IntRule==5
    % Gauss Laguerre [0 inf]-----
    for k=1:length(xGLa);
        int1(k) = wGLa(k)*HestonProb(xGLa(k),kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,1);
        int2(k) = wGLa(k)*HestonProb(xGLa(k),kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,2);
    end
    Npoints = length(xGLa);
    Domain = [xGLa(1) xGLa(end)];

```

FIGURE 11. Gauss-Laguerre Code

**3.6. Gauss-Legendre.** This quadrature is designed for the domain  $[-1, 1]$  so in order to be used, one must modified the Heston domain through the transformation:

$$(3.6.1) \quad \int_a^b f(x_0) dx = \frac{b-a}{2} \int_{-1}^1 f\left(\frac{b-a}{2}x + \frac{a+b}{2}\right) dx$$

In this approach the endpoints of the transformation (3.6.1),  $[a, b]$  are not included in the abscissas, allowing the user to set  $a = 0$  and  $b = x_N$ . The abscissas are the roots of the Legendre polynomial,  $P_N$ :

$$(3.6.2) \quad P_N(x) = \frac{1}{2^N} \sum_{k=0}^{\lfloor N/2 \rfloor} (-1)^k \frac{(2N-2k)!}{k!(N-k)!(N-2k)!}$$

where  $\lfloor \cdot \rfloor$  is the floor function. When  $N$  is even,  $P_N(x)$  contains only even powers of  $x$ , and vice-versa. The implementation of this method must ensure the inclusion of zeros where the polynomial does not have a power of  $x$ . For example if  $N$  is even, one must guaranteed that the odd powers are filled with zeros.

The G-Legendre weights depend on the derivative of the Legendre polynomial,  $P_N(x)$ . The weight function is defined as:

$$(3.6.3) \quad w_j = \frac{2}{(1-x_j^2)[P'_N(x_j)]^2}$$

It is worth emphasizing the fact that the domain transformation (3.6.1) must be correctly applied in order to properly implement this quadrature.

The code used in this study to implement the Gauss-Legendre numerical scheme 12 to calculate de Heston (1993) price is:

```
elseif IntRule==6
    % Gauss Legendre [-1 1]-----
    a=0;
    for k=1:length(xGLe);
        X      = (a+b)/2 + (b-a)/2*xGLe(k);
        int1(k) = wGLe(k)*HestonProb(X,kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,1)*(b-a)/2;
        int2(k) = wGLe(k)*HestonProb(X,kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,2)*(b-a)/2;
    end
```

FIGURE 12. Gauss-Legendre Code

**3.7. Gauss-Lobatto.** This last quadrature is also designed for the interval  $[-1, 1]$ , and therefore the domain of the Heston integrand must be modified like in the prior quadrature using transformation (3.6.1). This method uses the roots of the derivative of the Legendre polynomial of order  $N-1$  to calculate the abscissas. G-Lobatto uses the lower and upper bound of the integral as the first,  $x_1 = a$ , and last,  $x_N = b$ , abscissas. This is the main difference with G-Legendre.

The weight function of G-Lobatto use the derivative of the Legendre polynomials. The weight function is defined as:

$$(3.7.1) \quad w_j = \frac{2}{N(N-1)[P_{N-1}'(x_j)]^2}$$

for  $j=2,\dots,N$ . However, for the end points the weights must be defined by the following expression:

$$(3.7.2) \quad w_1 = w_N = \frac{2}{N(N-1)}$$

To implement this method one can not set  $a = 0$  because the first abscissa is  $x_1 = a$  and the CF is not defined at zero.

All the mentioned approaches in chapter 2, Heston and the improvement "Heston little trap" and consolidating the integral, were implemented with the seven numerical methods mentioned above, in order to calculate the Heston price.

The Heston vector parameters,  $\Theta = \{\kappa, \theta, \sigma, v_0, \rho\}$ , must be specified in order to completely describe and use the Heston model. The next two chapter explain how  $\Theta$  is found by two different methods: calibration by *MSE* and estimation by *MLE*. Those methods are two concepts that point to the same direction, one uses Q-measure in the sense of spot data, and the other P-measure in the sense of historical data.

The code used in this study to implement the Gauss-Lobatto numerical scheme 13 to calculate de Heston (1993) price is:

```
elseif IntRule==7
    % Gauss Lobatto [-1 1]-----
    for k=1:length(xGLo);
        X = (a+b)/2 + (b-a)/2*xGLo(k);
        int1(k) = wGLo(k)*HestonProb(X,kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,1)*(b-a)/2;
        int2(k) = wGLo(k)*HestonProb(X,kappa,theta,lambda,rho,sigma,T,K,S,rf,rd,v0,2)*(b-a)/2;
    end
    Npoints = length(xGLo);
end
```

FIGURE 13. Gauss-Lobatto Code

#### 4. OTC FX OPTION MARKET CONVENTION

In general, OTC market options like the one in Colombia, quote the option by delta rather than strike. This quotation method is common in OTC FX option markets, where buyers asks for a delta

(*i.e two months 25% $\Delta$ Call*) and the salesman or trader returns a price (*i.e 17,358%*)<sup>7</sup>, as well as the strike (*i.e 3200 COP*), given the spot reference. This means that for a given maturity, spot, and  $r_f, r_d$ , the market has a  $k$  that must be found, in order to make the formula (4.0.1) equal to either 25% or 10%, depending on the case.

$$(4.0.1) \quad \begin{aligned} \Delta &= e^{-r_f \tau} \mathcal{N}(d_1) \\ \text{with} \\ d_1 &= \frac{\log(S/K) + (r_d - r_f + \sigma^2/2)\tau}{\sigma\sqrt{\tau}} \end{aligned}$$

The strikes can be found by a root-finding algorithm like Newton-Raphson or bisection, nevertheless, it can also be found analytically. Equation (4.0.2) presents the analytical solution to go from  $\Delta$  to  $k$ .

$$(4.0.2) \quad K = S_0 \exp(-\mathcal{N}^{-1}(\Delta_f)\sigma\sqrt{\tau} + (r_d - r_f + \sigma^2/2)\tau)$$

where  $\Delta_f$  is the delta-forward. Given that the forward term has become relevant, the next two paragraphs will explain it.

There are four types of delta conventions used in the FX market, and the delta-forward is one of them.<sup>8</sup> In equity markets, the delta,  $\Delta$ , gives the amount of underlying the seller of the option must buy to hedge. In FX markets, that type of delta is called delta-spot,  $\Delta_S$ , and it is equivalent to buying  $\Delta_S$  times foreign units of the option's notional. The delta-forward,  $\Delta_f$ , is not only the derivative of the *BSM* option price with respect to the forward FX rate<sup>9</sup>,  $\frac{\partial C_{BSM}}{\partial f(t,T)} = \Delta_f = \mathcal{N}(d_1)$ , but also the number of forward contracts that the seller of the option needs to delta-hedge. See Beier and Renner (2010) [BR10] for a complete description of the standard FX market conventions worth reading.

Usually, traders use forward contracts to hedge options, and since the market prices in delta forward convention, the option hedge can be achieved much easier thanks to the quotations. For a better understanding, an example will be given: a trader sells a COP-USD option with maturity in two weeks for a given spot reference with a  $\Delta_{25}$  and a corresponding strike of 3200 COP. Once sold, the trader must hedge the

<sup>7</sup>The FX prices are often in terms of volatility or pips, not in currencies like COP or USD.

<sup>8</sup>Recall that delta,  $\Delta$  measures the rate of change of the price with respect to the underlying,  $\Delta = \partial C / \partial S$  where C stands for call price under *BSM*

<sup>9</sup>Express the *BSM* formula for FX market as  $C = e^{r_d \tau} [f(t,T)\mathcal{N}(d_1) - K\mathcal{N}(d_2)]$  where  $d_1 = \frac{\ln(f(t,T)/K) + (\sigma^2/2)\tau}{\sigma\sqrt{\tau}}$ ,  $d_2 = d_1 - \sigma\sqrt{\tau}$  and  $f(t,T) = S_t e^{r_d - r_f \tau}$ .

option. He needs to make the replicating portfolio or get in a forward contract with the same maturity and strike of the option. The chosen hedge is the forward contract, since the market already possesses the forward with the needed characteristic.

To summarize delta conventions, they express the strikes in terms of a *BSM* greek: delta can refer to spot or forward, amongst others, where spot means the delta hedge must be made in the spot market, and forward in the forward one. ATM convention will be presented in the next paragraph.

The ATM-forward means that the strike is not equal to the spot, instead it is equal to the forward for the given maturity. This implies that by the put-call parity,  $Call - Put = e^{rd\tau}(F(t, T) - K)$ , this is the strike at which the price of the call and put are the same. There is also a put call parity for deltas:  $\Delta_C - \Delta_P = 1$ , that will be helpful since it means that  $10\Delta Put$  is equal to  $90\Delta Call$  and  $25\Delta Put$  to  $75\Delta Call$ . This is useful because in the Colombian OTC FX option market, the most traded deltas are  $10\Delta Put$ ,  $25\Delta Put$ , *ATM*,  $25\Delta Call$ ,  $10\Delta Call$  and they can all be expressed in terms of  $\Delta_C$  thanks to the put-call parity's presented. For each delta, one can find  $1W$ ,  $1M$ ,  $2M$ ,  $3M$ ,  $6M$ ,  $9M$ , and  $1Y$  as maturities.

## 5. ESTIMATION: MAXIMUM LIKELIHOOD

This study uses the Atiya and Wall (2009) analytic approximation for the likelihood function of the Heston model. Recall that the Heston model does not define the volatility as a function of past asset observations, instead, it defines it as a latent variable state in a stochastic process.

Because the underlying distribution is not needed to price in the Heston model, it is not defined in the specifications, and therefore the classical construction of the likelihood function can not be achieved.

Atiya and Wall (2009) suppose that the transition probability density for the joint log-underlying price/variance process from  $t$  to  $t + 1$  is bivariate normal, where  $\mathcal{N}$  denotes the normal density of mean  $\mu_{t+1}$  and covariance matrix  $\Sigma_{t+1}$ . The following equation illustrates the supposed bivariate normal distribution:

$$(5.0.1) \quad p(S_{t+1}, v_{t+1} | S_t, v_t) = \mathcal{N}(\boldsymbol{\mu}_{t+1}, \boldsymbol{\Sigma}_{t+1})$$

To check the entire specifications of  $\boldsymbol{\mu}_{t+1}$  and  $\boldsymbol{\Sigma}_{t+1}$  see Atiya and Wall (2009).

Atiya and Wall (2009) defined the problem of estimating the volatility as a "filtering problem" and formulated it as that of obtaining the likelihood of the volatility given the past likelihood observations. In other words, the variance likelihood must be approximated from the underlying one. The likelihood at time  $t + 1$  is:

$$(5.0.2) \quad L_{t+1}(v_{t+1}) \propto d_t (ab_t)^{-1/4} e^{-2\sqrt{ab_t}} L_t(v_t)$$

the equation (5.0.2) is only approximate but it is valid because the approximation error is small. This is due because the time step is short. That implies that the gap between the constant function and the continuous one are small. Please see Atiya and Wall (2009) for more details.

Once the likelihood is defined, calculating the log-likelihood is trivial. The above equation uses the following quantities:

$$(5.0.3) \quad \begin{aligned} a &= \frac{(\kappa')^2 + \rho\sigma\kappa' dt + \sigma^2(dt)^2/4}{2\sigma^2(1 - \rho^2)dt} \\ b_t &= \frac{(v_{t+1} - \alpha dt)^2 - 2\rho\sigma(v_{t+1} - \alpha dt)(\Delta x_{t+1} - \mu dt) + \sigma^2(\Delta x_{t+1} - \mu dt)^2}{2\sigma^2(1 - \rho^2)dt} \\ d_t &= \frac{1}{D} \exp\left(\frac{(2\kappa' + \rho\sigma dt)(v_{t+1} - \alpha dt) - (2\rho\sigma\kappa' + \sigma^2 dt)(\Delta x_{t+1} - \mu dt)}{2\sigma^2(1 - \rho^2)dt}\right) \end{aligned}$$

with  $\kappa' = 1 - \kappa dt$ ,  $\alpha = \kappa\theta$ ,  $D = 2\pi\sigma\sqrt{1 - \rho^2 dt}$ , initial values<sup>10</sup>  $L_0(v_0) = e^{-v_0}$ , the drift  $\mu = r_d - r_f$ , and the log-underlying increments  $\Delta x_{t+1} = x_{t+1} - x_t$ .

Atiya and Wall (2009) proposed a value for the time increment of  $dt = 1/252$  for daily data.

Note that  $b_t$ ,  $d_t$  and also  $L_{t+1}(v_{t+1})$  depend on  $v_t$ , and therefore, to compute the likelihood,  $v_t$  must be calculated first. Atiya and Wall (2009) notice that  $v_t = \sqrt{b_t/a}$  and invert it in order to get:

---

<sup>10</sup>Here  $v_0$  is the initial variance parameter.

$$(5.0.4) \quad v_{t+1} = \sqrt{B^2 - C} - B$$

where

$$\begin{aligned} B &= -\alpha dt - \rho\sigma(\Delta x_{t+1} - \mu dt), \\ C &= (\alpha dt)^2 + 2\rho\sigma\alpha dt(\Delta x_{t+1} - \mu dt) + \sigma^2(\Delta x_{t+1} - \mu dt)^2 - 2v_t^2 a \sigma^2(a - \rho^2)dt. \end{aligned}$$

To implement the estimation one must construct the likelihood using the following steps. First, calculate the quantities that do not change in the likelihood construction such as  $\alpha$ ,  $a$ ,  $D$  and the initial value  $L(v_0)$ . Second, calculate  $\Delta x_{t+1}$  in order to find  $B$  and  $C$ , which are needed to obtain  $v_{t+1}$ . Third, Atiya and Wall (2009) state, "it is imperative to combine the exponent in  $d_t$  and the exponent  $-2\sqrt{ab_t}$  before exponentiating". Doing this will avoid numerical errors. The reason is that there are some exponents that are large and almost of equal magnitude but with opposite sign. Fourth, once  $d_t$  and  $e^{-2\sqrt{ab_t}}$  are computed as one, the remaining terms in (5.0.2) should be multiplied. Finally, a for-loop from step 2 until 4 for  $t = 0$  to  $T - 1$  must be made to obtain the likelihood. Computing the log-likelihood is straightforward once the above steps are understood.

The code used in this study to implement the Likelihood Atiya and Wall (2009) estimation is 14.

It is worth mentioning that this study define  $v_0$  as a parameter and do not uses a grid to define it.

The parameter found using the proposed data, initial values and MLE estimation method lead to: 15

## 6. CALIBRATION: NON-LINEAR LEAST SQUARE/ MEAN SQUARE ERROR

Calibration is a method, where a few parameters from the model are tweaked until they match with their counterpart market values, so that the model may fit with the market data. This is done by minimizing the square distance between the model and market prices, which is known as *MSE*. One can also minimize the model's implied volatility with the market one, as long as they both have the same dimensions.

This method calibrates the Heston vector parameter  $\Theta = \{\kappa, \theta, \sigma, v_0, \rho\}$  by defining a function that minimize the square distance between the



```

function [y,v] = LikelihoodAW(param,x,r,q,dt,method)

% INPUTS %

% Name the Heston parameters
kappa = param(1);
theta = param(2);
sigma = param(3);
v0 = param(4);
rho = param(5);

% Atiya and Wall parameterization
alpha = kappa*theta;
beta = kappa;

% Number of log-stock prices
T = length(x);

% Drift term
mu = r - q;

% Equation (17)
betap = 1 - beta*dt;

% Equation (18) - denominator of d(t)
D = 2*pi*sigma*sqrt(1-rho^2)*dt;

% Equation (14)
a = (betap^2 + rho*sigma*betap*dt + sigma^2*dt^2/4) / (2*sigma^2*(1-rho^2)*dt);

% Variance and likelihood at time t = 0
v(1) = v0;
% Variance and likelihood at time t = 0
v(1) = v0;
if method==1
    L(1) = exp(-v(1)); % Construct the Likelihood
elseif method==2
    L(1) = -v(1); % Construct the log-likelihood
end

% Construction the likelihood for time t = 1 through t = T
for t=1:T-1
    % Stock price increment
    dx = x(t+1) - x(t);
    % Equations (31) and (32)
    B = -alpha*dt - rho*sigma*(dx-mu*dt);
    C = alpha^2*dt^2 + 2*rho*sigma*alpha*dt*(dx-mu*dt) + sigma^2*(dx-mu*dt)^2 - 2*v(t)^2*a*sigma^2*(1-rho^2)*dt;
    % Equation (30) to update the variance
    if B^2 - C > 0
        v(t+1) = sqrt(B^2 - C) - B;
    else
        % If v(t+1) is imaginary use the approximation Equation (33)
        bt = ((v(t)-alpha*dt)^2 - 2*rho*sigma*(v(t)-alpha*dt)*(dx-mu*dt) + sigma^2*(dx-mu*dt)^2) / (2*sigma^2*(1-rho^2)*dt);
        if bt/a > 0
            % Equation (33)
            v(t+1) = sqrt(bt/a);
        else
            % If v(t+1) is still negative, take the previous value
            v(t+1) = v(t);
        end
    end
    % Equation (15) and (16)
    bt = ((v(t+1)-alpha*dt)^2 - 2*rho*sigma*(v(t+1)-alpha*dt)*(dx-mu*dt) + sigma^2*(dx-mu*dt)^2) / (2*sigma^2*(1-rho^2)*dt);
    x1 = ((2*betap+rho*sigma*dt)*(v(t+1)-alpha*dt) - (2*rho*sigma*betap+sigma^2*dt)*(dx-mu*dt)) / (2*sigma^2*(1-rho^2)*dt);
    x2 = -2*sqrt(a*bt);
    % Combined exponent for Equation (34)
    E = exp(x1 + x2) / D;
    if method==1
        % Equation (34) for the likelihood L(t+1)
        L(t+1) = (a*bt)^(-1/4) * E * L(t);
    elseif method==2
        % Alternatively, use the log-likelihood, log of Equation (34)
        L(t+1) = -1/4*log(a*bt) + x1 + x2 -log(D) + L(t);
    end
end

% Negative likelihood is the last term.
% Since we maximize the likelihood, we minimize the negative likelihood.
y = -real(L(T));

```

FIGURE 14. Likelihood AW Code

Maturity	Atiya-Wall MLE Estimates				
	kappa	theta	sigma	v0	rho
T = 7 days	19.9999	0.0000	1.8138	0.0339	0.7559
T = 30 days	19.9997	0.0751	0.7984	0.0425	0.9989
T = 60 days	11.9788	0.0425	0.0003	0.0309	0.2728
T = 90 days	9.0277	0.0364	0.0022	0.0438	0.9977
T = 180 days	19.9992	0.0411	0.0001	0.0247	0.6482
T = 270 days	8.8073	0.0435	0.0000	0.0227	0.9979
T = 365 days	11.7398	0.0414	0.0000	0.0264	0.9966

FIGURE 15. MLE Estimator

spot data and the model one. The minimizing parameters found are the ones used. This implies that the market and the Heston prices are as similar as can be in square error. As explained before, the calibration method finds the parameters under the risk-neutral measure,  $Q$ .

The function can be defined in one of two ways: as  $MSE$  or as  $IVMSE$ . The first, Mean Square Error or  $MSE$ , is defined as:

$$(6.0.1) \quad \frac{1}{N} \sum_{t,k(\Delta)} w_{t,k(\Delta)} (C_{t,k(\Delta)}^{mkt} - C_{t,k(\Delta)}^H)^2$$

This function minimizes the square error between the market and Heston prices, where the subindex refers to all the possible combinations of the Colombian liquidity points: seven maturities and the five strikes (in terms of 10 and 25 deltas).

To apply the  $MSE$  function, one must retrieve the prices from the market data which are in terms of delta-forward and an ATM-forward. The last two terms are quoted normally in OTC (over-the-counter) FX option markets and they must be understood somehow in order to retrieve the prices.

In brief, to apply the  $MSE$  function with the Colombian market data: deltas<sup>11</sup> the market implied volatility expressed in combination of  $90\Delta$  Call,  $75\Delta$  Call, ATM,  $25\Delta$  Call,  $10\Delta$  Call, and maturities 1W, 1M, 2M, 3M, 6M, 9M, 1Y, must be used with equation (4.0.2)

<sup>11</sup> The  $\Delta_{Put}$  were changed for  $\Delta_{Call}$

to find the respective strikes for the given deltas:  $\text{MktIV} \rightarrow \text{Stike}$

followed by this apply *BSM* pricing formula to retrieve the prices of the european vanilla call for all the combination of maturities and deltas:  $\text{Stike} \rightarrow \text{BSM Price}$ . This means that one part of (6.0.1) is know available to use :  $C_{t,k(\Delta)}^{mkt}$

In addition,  $C_{t,k(\Delta)}^H$ )<sup>2</sup> must be calculated so equation (4.0.2) can be used. For calibration purpose this study calculate the Heston price by Heston (1993) with Gauss Laguerre.

The main disadvantage of the *MSE* is that short maturities, deep OTM with little value, do not contribute enough to the sum in (6.0.1). Hence, the optimization will tend to fits better ITM options with longer maturities. The weights used in this study for the *MSE* function are equal for all data. In order to solve the mentioned problem the end user can change the weights by assigning a relative big weight to shorter maturities deep OTM.

The *MSE* function can also be defined with the implied volatility of the market. The Implied Volatility Mean Square Error, *IVMSE*, is the second way of defining the function.

$$(6.0.2) \quad \frac{1}{N} \sum_{t,k(\Delta)} w_{t,k(\Delta)} (IV_{t,k(\Delta)}^{mkt} - IV_{t,k(\Delta)}^H)^2$$

The *IVMSE* finds the parameter set, so that the implied volatilities of the model are as close as possible to those of the market. Implementing this function to calibrate the Heston model was done by following the following steps: express the deltas in terms of strikes for all maturities, compute the Heston prices with the given strikes and maturities, use a root-finding algorithm to find the *BSM* volatility that equal the Heston price with the market price, and finally, add up the square differences.

The main disadvantage of *IVMSE* is the need of a root-finding algorithm which is numerically intensive. One remedy is to use the approximation of the implied volatility that the Vol. of Vol. expansion series given by Lewis (2000), (A.3.4b). Another solution is to approximate the *IVMSE* by the function given in Christoffersen et al. (2009). See [CHJ09]. The parameter set estimated from the last mentioned method minimize the following function:

$$(6.0.3) \quad \frac{1}{N} \sum_{t,k(\Delta)} \frac{(IV_{t,k(\Delta)}^{mkt} - IV_{t,k(\Delta)}^H)^2}{Vega_{t,k(\Delta)}^2}$$

were  $Vega_{t,k(\Delta)}^2 = S_0 \mathcal{N}(d1) \sqrt{\tau}$  is the *BSM* price equation sensitivity with respect to the implied volatility evaluated at the maturity and strikes. This function offers a reduced computational time, however, it is done at the expense of a loss of accuracy. Christoffersen et al. (2009) states that the same function used to calculate the parameters must be used to evaluate the model fit.

The code used in this study to implement the Non-Linear Least Square calibration 16 is:

There are considerable philosophical discrepancies between calibration and estimation. Quants may argue that under the risk-neutral distribution  $Q$ , the prices are arbitrage-free, but under  $P$  – *measure* they are not. Econometricians may stand for a more robust method to find the parameters set. They may argue that historical prices carry information about the expected value of the price itself and therefore the parameters are arbitrage-free. Nevertheless, there is a similarity in applying any of the methodologies, both algorithms may find the zeros in the functions, maximize or minimize in order to find the parameter set.

## 7. AN ACCURATE IMPLEMENTATION

This study calculates the Heston price with two equivalent pricing approaches: (i) Heston (1993) and (ii) Consolidating the integrals

Because the optimization method used in this study depends on the Strike rather than Delta, one must find the Strike for the given Deltas. Furthermore, one can not assume there is a linear relationship between the Delta and the Strike even though a one to one or injective function must exist between them. In other words, a delta can only have one strike and viceversa, and that is a necessary and sufficient condition to proceed. The following graph shows the relationship.<sup>17</sup>

In order to implement the Heston model, the parameter set must be specified first. This study uses the Matlab function `fmincon` to minimize the *NLLS* and *MLE*. These optimization functions are stable and is highly accepted in different disciplines. The constraints used

```

function y = NL_Least_Square(param,S,rd,rf,MktPrice,K,T,PutCall,MktIV,ObjFun,Method,a,b,Tol,MaxIter,N,xGLa,wGLa)
%The most popular way to estimate the parameters of the Heston model is Non-linear %...%

%Parameters
kappa = param(1);
theta = param(2);
sigma = param(3);
v0 = param(4);
rho = param(5);
lambda = 0;

[N,T,NK] = size(MktPrice);

for k=1:NK
    % Select the method for obtaining the price
    if Method==1
        % Heston Call price with Gauss Laguerre integration
        CallPrice = HestonPriceGaussLaguerre('C',S,K(k),T,rd,rf,param,xGLa,wGLa);
    elseif Method==2
        % Lewis vol-of-vol Series II expansion Call price
        [iv, CallPrice] = SeriesIICall(S,K(k),rd,rf,T,v0,rho,theta,kappa,sigma);
    elseif Method==3
        % Lewis vol-of-vol Series I expansion Call price
        CallPrice = SeriesICall(S,K(k),rd,rf,T,v0,rho,theta,kappa,sigma);
    elseif Method==4
        % Mid-Point
        CallPrice = Price93('C',S,K(k),T,rd,rf,kappa,theta,sigma,lambda,v0,rho,1,a,b,N,xGLa,wGLa,xGI);
    elseif Method==5
        % Trapezoidal Rule
        CallPrice = Price93('C',S,K(k),T,rd,rf,kappa,theta,sigma,lambda,v0,rho,2,a,b,N,xGLa,wGLa,xGI);
    elseif Method==6
        % Obtain the call price or put price
        if PutCall(k)=='C'
            ModelPrice(k) = CallPrice;
        else
            ModelPrice(k) = CallPrice - S*exp(-rf*T) + exp(-rd*T)*K(k);
        end

        % Select the objective function
        if ObjFun == 1
            % MSE
            error(k) = (MktPrice(k) - ModelPrice(k))^2;
        elseif ObjFun == 2
            % RMSE
            error(k) = (MktPrice(k) - ModelPrice(k))^2 / MktPrice(k);
        elseif ObjFun == 3
            % IVMSE
            if Method==2
                ModelIV(k) = iv;
            else
                ModelIV(k) = BiseCBSIV(PutCall(k),S,K(k),rd,rf,T,a,b,ModelPrice(k),Tol,MaxIter);
            end
            error(k) = (ModelIV(k) - MktIV(k))^2;
        elseif ObjFun == 4
            % IVMSE Christoffersen, Heston, Jacobs proxy
            d = (log(S/K(k)) + (rd-rf+MktIV(k)^2/2)*T)/MktIV(k)/sqrt(T);
            Vega(k) = S*normpdf(d)*sqrt(T);
            error(k) = (MktPrice(k) - ModelPrice(k))^2 / Vega(k)^2;
        end
    end
end

y = sum(error) / (NK);

```

FIGURE 16. Non-Linear Least Square Code

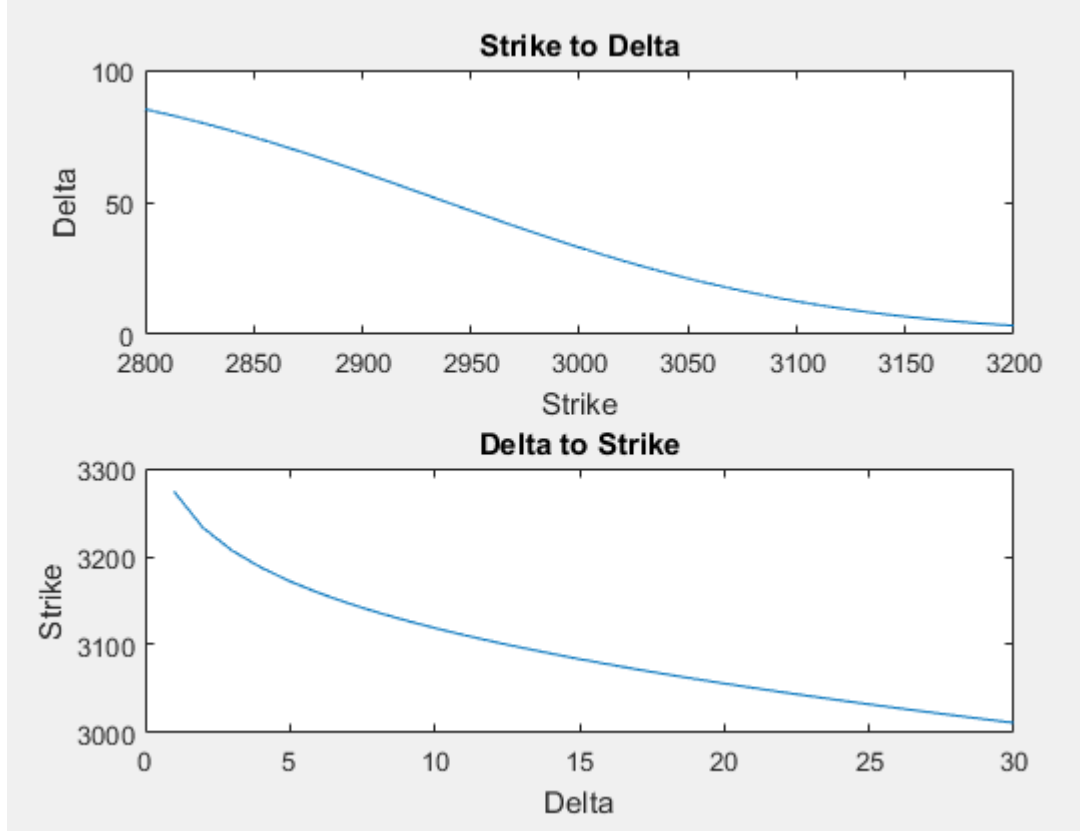


FIGURE 17. Delta to Strike and Strike to Delta

are  $\kappa > 0$ ,  $\theta > 0$ ,  $\sigma > 0$ ,  $v_0 > 0$ ,  $\rho \in [-1, 1]$ , starting values  $\kappa = 8$ ,  $\theta = 0.04$ ,  $\sigma = 0.3$ ,  $V_0 = 0.05$ ,  $\rho = 0.8$  and some reasonable lower and upper boundaries. Moreover, due to the optimization method used, one must define an upper boundary for all parameters. The code snippet illustrated in figure 18 shows the initial values, and the lower and upper boundaries used in this study.

it is worth mentioning that  $\sigma$  and  $V_0$  are the volatility of volatility and the initial value of the volatility of volatility.

It is worth mention that initial values were hardly found. The Colombian market requires a positive  $\rho$  and the goal was achieved by applying a try and failure method in a neighborhood defined by the lower and upper bound.

```

% Bounds on the parameter estimates and starting values
e = 1e-5;
%      kappa theta  sigma v0    rho
lb  = [e      e      e      e    -0.999]; % Lower bound on the estimates
ub  = [20     2      2      3     0.999]; % Upper bound on the estimates
start = [8.0  0.04  0.3  0.05  0.8];

```

FIGURE 18. Initial values, lower and upper bound

As with any optimization, it is important that the starting and true values do not lie too far from each other<sup>12</sup>. To achieve that, one must be careful using empirical literature to define them because almost all of the reliable studies use index or stock data. For FX emerging markets data such as USD-COP, the  $\rho$  must be positive to ensure the market behavior: If the risk increases the investors will find shelter in the strong pair leading to price increases. In other words, the literature may not be helpful in defining the starting values. In order to define them one must focus on Fellers condition,  $2\kappa * \theta > \sigma^2$ , to ensure that  $\sigma$  is always positive. In the one week parameter set the condition is not met due to the Stochastic model's well known limitation. Zhu (2009) [Zhu09] calibrates  $V_0$  to the ATM implied volatility in the FX market, and suggest setting  $\kappa$  large enough so that Fellers condition is fulfilled.

In the FX literature the emerging market is not as popular as desired, because of the liquidity in the market. This implies that finding initial values to used defining the parameter set is one of the paper contributions. The author remark in the difficult to find initial values that lies to a positive correlation because parameter set was estimate for each maturity. This implies that the values used are stable for the Colombian FX market.

After the problem of the initial values parameter is solved, one can proceed to estimate the parameters used to find the Heston prices. This study estimates a set of parameters for each maturity due to the recommendation of [RH13]. The main difference in each estimation is that the farther the maturity, the more data is used to estimate the set parameter. For example, in the estimation in the one week parameter set, only data from the previous week is considered, while for the one month parameter set, only data from the previous month is considered, and so on. This is significantly important due to the fact that

---

<sup>12</sup>This is a Ansatz: Suppose this is true so that one can proceed.

```

for t=1:NT
    % Historical stock prices. Use D days before 12 jul 2016.
    D(t) = T(t)*365; %#ok<SAGROW>
    S = xlsread('Mtk data 12 jul 2016.xlsx', 'Spot', ...
        ['B2:B' num2str(D(t)+1)]);
    % Put oldest prices first and calculate log prices
    S = flipud(S);
    x = log(S);
    % Obtain the parameters estimates using Atiya and Wall MLE
    [MLE_Parameter(t,:)]= fmincon(@(p) LikelihoodAW(p,x,...
        rd(t),rf(t),dt,Lmethod),start,[],[],[],[],lb,ub,[],options);

    % Obtain the parameter estimates using option prices and loss
    [NLLS_Parameter(t,:)] = fmincon(@(p) NL_Least_Square(p,Spot,...
        rd(t),rf(t),MktPrice(t,:),Strike(t,:),T(t),PutCall,...
        MktIV(t,:),ObjFun,Method,a,b,Tol,MaxIter,N,xGLa,wGLa,...
        xGLe,wGLE,xGLo,wGLo),start,[],[],[],[],lb,ub,[],options);
end

```

FIGURE 19. Calibration snippets code

even though the same data may be used on multiple calculations, the amount of data used varies from calculation to calculation. The same parameter accross every maturity must not lie too far from each other. The Matlab snippet of code used to calculate the parameters is shown in figure 19

The chosen non-linear least square function was  $MSE$  because [RH13] suggests to use it. Once the parameters are found by the chosen method one can proceed. It is worth mentioning that for the optimization method to work, one must define a function to find the price of the Heston model. In other words, the optimization method uses a pre-selected Heston pricing equation, and the resulting parameters set will be used to find the most accurate Heston price method. So the same variable first will be endogenous and later it becomes exogenous. This can be misleading, however, one may think that the pre-selected pricing method can be skewed, but that is never the case. No matter what pre-selected pricing method the code uses, it is never the most accurate.

Once the estimation problem is overcome, one should find, in one hand, the more accurate method and, in the other, the computationally faster scheme, and combine both results to find the answer this



Estimates	kappa	theta	sigma	v0	rho	Hessian
-----						
T = 7 days						
Atiya-Wall MLE Estimates	19.9999	0.0000	1.8138	0.0339	0.7559	16.4219
Non-Linear Least Square	0.6714	1.8529	1.9999	0.0200	0.1479	2.0000
-----						
T = 30 days						
Atiya-Wall MLE Estimates	19.9997	0.0751	0.7984	0.0425	0.9989	68.6490
Non-Linear Least Square	0.3934	1.9114	1.0674	0.0002	0.3000	2.0000
-----						
T = 60 days						
Atiya-Wall MLE Estimates	11.9788	0.0425	0.0003	0.0309	0.2728	144.5870
Non-Linear Least Square	0.1747	1.9523	0.7321	0.0002	0.3462	2.0000
-----						
T = 90 days						
Atiya-Wall MLE Estimates	9.0277	0.0364	0.0022	0.0438	0.9977	216.4523
Non-Linear Least Square	0.1144	1.9097	0.6624	0.0004	0.3621	2.0000
-----						
T = 180 days						
Atiya-Wall MLE Estimates	19.9992	0.0411	0.0001	0.0247	0.6482	441.8461
Non-Linear Least Square	0.0651	1.6204	0.5224	0.0004	0.3549	2.0000
-----						
T = 270 days						
Atiya-Wall MLE Estimates	8.8073	0.0435	0.0000	0.0227	0.9979	574.7408
Non-Linear Least Square	0.2618	0.2667	0.4620	0.0001	0.3702	2.0000
-----						
T = 365 days						
Atiya-Wall MLE Estimates	11.7398	0.0414	0.0000	0.0264	0.9966	577.7421
Non-Linear Least Square	1.4763	0.0477	0.5520	0.0011	0.3699	2.0000
-----						

FIGURE 20. MLE Vs. NLLE

study is looking for: a scheme that lasts the least amount of time possible and a pricing method that shows as small an error as possible. In order to achieve this goal, a Ceteris Paribus analysis will take place, first finding the computationally Faster Scheme, and then the most accurate method.

In the next chapter the problem of the computationally faster scheme will take place.

## 8. COMPUTATIONALLY FASTER SCHEME

In order to make the process of finding the computationally faster solution, all the possible combinations of the pricing approach that require to solve an integral with all the numerical schemes were computed one hundred times. In other words, the two integrals of Heston (1993) were calculated with the 7 numerical schemes. After the code calculate the Heston (1993) price by 7 different ways, one must know which of those 7 ways is the best. Understand the best as the scheme

that lasts the least amount of time possible and show the small error as possible. Using the Ceteris Paribus analysis the problem become two: (i) the accurest method and (ii) the fastes one.

To solve the problem of the fastest numerical scheme one must measure the time required to run the code. This study uses the Matlab function Timeit. This function calls the specified function multiple times, and returns the median of the measurements to form a reasonably robust time estimate. It also considers first-time costs. The Matlab function cputime was discarded because it could be misleading. For further details on measure performance se <http://www.mathworks.com/>

This study uses more than one repetition to make more robust the analysis because if the repetition are not enough, the conclusion of the study will be difficult to accept by the reader. To solve this problem, for each pricing method the code calculates an array of:

$$(repetition, Strike, Maturity, IntegrationScheme)$$

were: repetition=100, Strike=ATM, Maturity=[1W,2W,3W,1M,6M,9M,1Y], Integration Scheme=(i)Mid-Point (ii) Trapezoidal rule, (iii) Simpson's rule, (iv) Simpson's 3/8 rule, (v) Gauss-Laguerre, (vi) Gauss-Legendre and (vii) Gauss-Lobato. In other word the array will have dimention:

$$(100, 1, 7, 7)$$

Figure 29 shows the array for a combination of a pricing aproach and scheme. There are 49 combinations of pricing methods and numerical schemes. For example:

$$(rep = 1, Strike = ATM, Maturit = 1W, IntegrationScheme = Mid-Point)$$

$$(rep = 2, Strike = ATM, Maturit = 1W, IntegrationScheme = Mid-Point)$$

$$(rep = 3, Strike = ATM, Maturit = 1W, IntegrationScheme = Mid-Point)$$

.

.

.

$$(rep = 100, Strike = ATM, Maturit = 1W, IntegrationScheme = Mid-Point)$$

is one combination, of Strike, Maturity and the integration scheme and the code sees it as a matrix. Particular, all the 49 combination the code sees it as an array. The code uses nested loops to calculate all the combination. ie. First make the 100 rep, of ATM strike and 1W maturity, then its go to 100 rep, ATM strike and 1M maturity. After that its go to 100 rep of ATM strike and maturity of 2M, so on, until the maturity of 1Y. See Figure 29 and imagine the Array of:

$$(repetition, Strike, Maturity)$$

Now think that in that array lies data that use only mid-point numerical scheme to find those 700 price: 7 maturities, and 100 rep for each

The study calculate the 29 array to all the numerical scheme. That mean that will be calculated 7 times. Imagine 30 as the form the code calculate 100 rep, with the ATM strike, accross all the maturitis, and solving the model integral with all the numerical shcheme this study analice. It is important that the reader remeber that the integrand change when the parameter of the model change. In this study the parametes change every time the maturity change due to the fack that we calculate a fiderent set of parameters to each maturity, and that lies in important change in the integrand when the maturity change.

All of the above leads to an array dimension problem. To solve this, one must calculate the average smile time for each maturity and for all the posible combinations of pricing methods and numerical schemes to get an array of dimention

$$(repetition = 100, Strike = 1, Maturity = 7, IntegrationScheme = 2) = (100, 1, 7, 2)$$

See Figure 21 and imagine the array.

Once the code calcuates the mentioned array one can reduce the dimention to see the computationally faster scheme. Rember thar for eache calculation of price the code save the time it take to calculate that price. The next challenge is to find another array that reflects the

	Mid-Point	Trapezoidal rule	Simpson's rule	Simpson's 3/8 rule	Gauss-Laguerre	Gauss-Legendre	Gauss-Lobato
Heston (1993)	Time 1,1	Time 1,2	Time 1,3	Time 1,4	Time 1,5	Time 1,6	Time 1,7
Consolidating the integrals	Time 2,1	Time 2,2	Time 2,3	Time 2,4	Time 2,5	Time 2,6	Time 2,7
Carr and Madan (1997)	Time 3,1	Time 3,2	Time 3,3	Time 3,4	Time 3,5	Time 3,6	Time 3,7
C1 and (v) C2 fundamental transform	Time 4,1	Time 4,2	Time 4,3	Time 4,4	Time 4,5	Time 4,6	Time 4,7
volatility of volatility series expansion C1	Time 5,1	Time 5,2	Time 5,3	Time 5,4	Time 5,5	Time 5,6	Time 5,7
volatility of volatility series expansion C2	Time 6,1	Time 6,2	Time 6,3	Time 6,4	Time 6,5	Time 6,6	Time 6,7

FIGURE 21. Tiempo ATM

most accurate pricing method.

## 9. ACCURATE METHOD

It may be assumed that the trapezoidal rule is the most accurate pricing approach since a curved line can be approximated by joining straight lines. For example, a circle can be defined as a polygon of infinite sides. Therefore, one can accurately calculate the Heston integral by applying the trapezoidal rule with the lower boundary as close as possible to zero and an upper limit greater than the point where the integral tends to zero. For the lower limit the study uses  $1e-20000$ , an upper limit equal to 300 and divide that space in 10.000.000. The Figure 22 , 23 ,24 and 25 show the integrals of Heston 1993 for maturity of 1 week and 1 year using MSE and NLLS

All the integrals were plotted to visually determine the point where the integral tends to zero, which resulted in the conclusion of using 300 as the upper limit. This does not mean that the integrals in the Colombian market decay near that number. However, it can be ensured that the point of decay is less than the chosen upper limit.

In order to approximate a curved line that can be steep at the origin but highly oscillatory, the trapezoidal rule must divide the integration interval into a very large number of parts. This study divides the chosen integration domain  $[a, b]$  in 10 million parts. The drawback with this implementation is the extremely high computational time it can take to process this numerical method. The benchmark prices for all the maturities and deltas have at least 41 decimal points, see figure 26, and take 1.5 hours to calculate each price.

Once the benchmark price has been determined, the MSE between the benchmark and all the possible combinations of price methods and

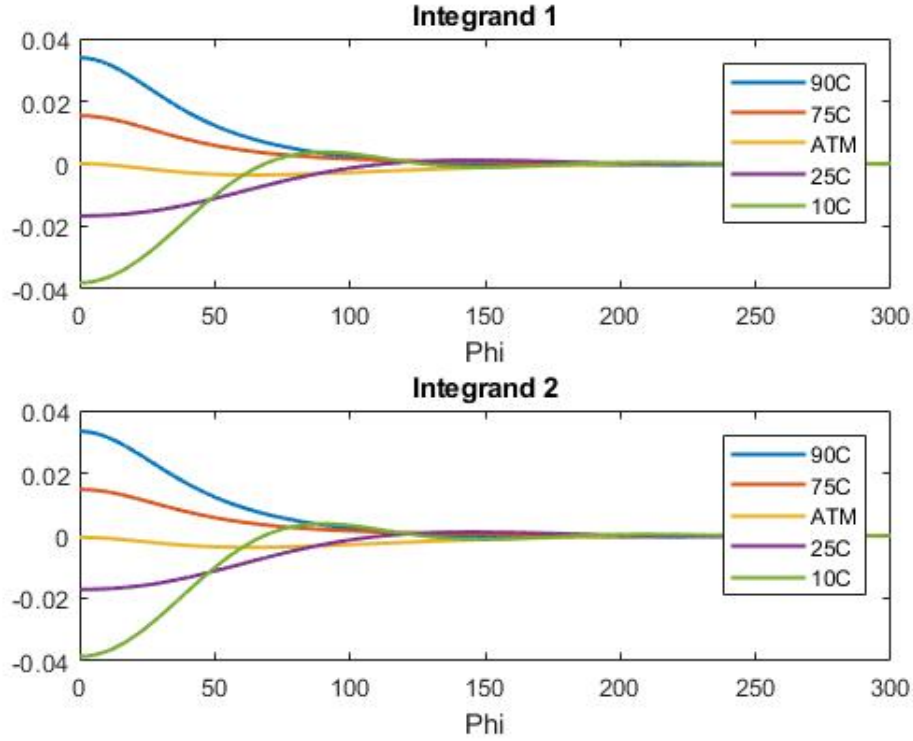


FIGURE 22. 1W Benchmark Integral using MLE parameters

schemes must be calculated, in order to finally compute the average of a given MSE price method across maturities. At the end one will have an MSE 7x7 matrix. See 27.

## 10. JOINING THE AVERAGE TIME WITH THE MSE

For each price method, a Time versus MSE plot is drawn after each of the axis variables are calculated. Based on these plots, one can proceed to a more detailed analysis by dividing each plot into four regions. An example of how this division should take place is shown in Figure 28. From the regions in the graph, the following conclusions can be derived:

- Region one is the goal behind this study, aiming at finding the most perfect balance between MSE and computational time for each pricing method.
- Region two and three vary in importance based on what the code user is looking for. Where region two would appeal to those who prefer to sacrifice more time in exchange for a more

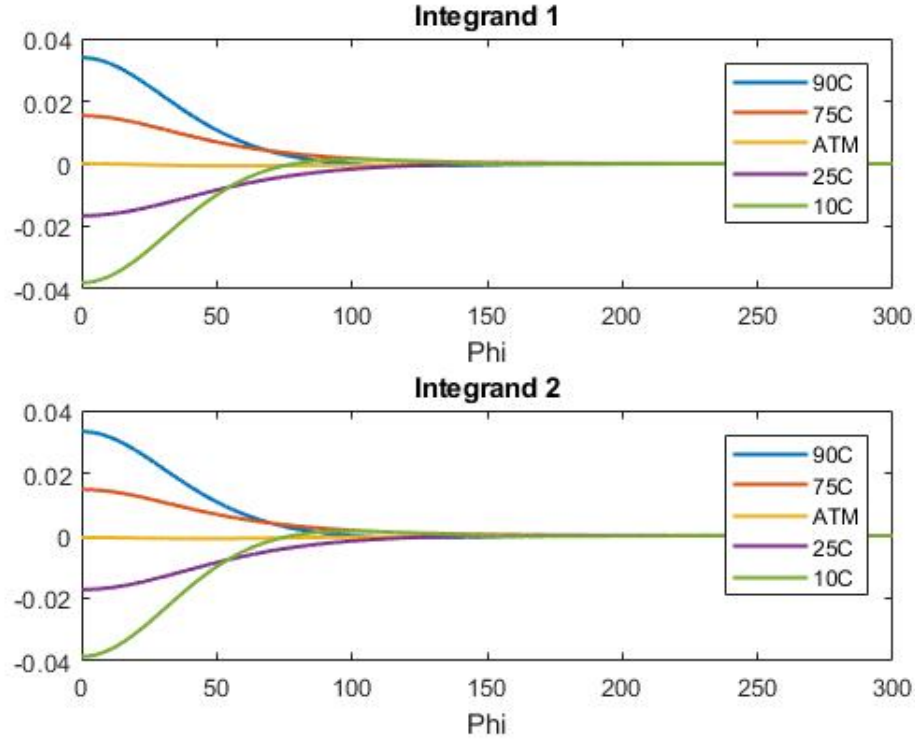


FIGURE 23. 1W Benchmark Integral using NLLS parameters

accurate result, while region three would favor those with less time to spend.

- Region four reflects the most undesirable ways to calculate the Heston price.

In an ideal scenario, there would be six graphs, one for each pricing method. This would allow for the creation of a final graph displaying only the top result from each pricing method. With this last graph, the best pricing method and numerical scheme would be selected.

It is worth mentioning that if any of the numerical calculations of the price, regardless of numerical schemes and price methods used, lead to a negative value, the price method will not be taken into account. The reason behind this lies in the fact that if a single price is negative, there is a significant probability that other calculations using the same method can yield more undesirable results. This does not imply that if all the results are positive, other calculations using that same pricing method will not be negative, but the chances of calculating a

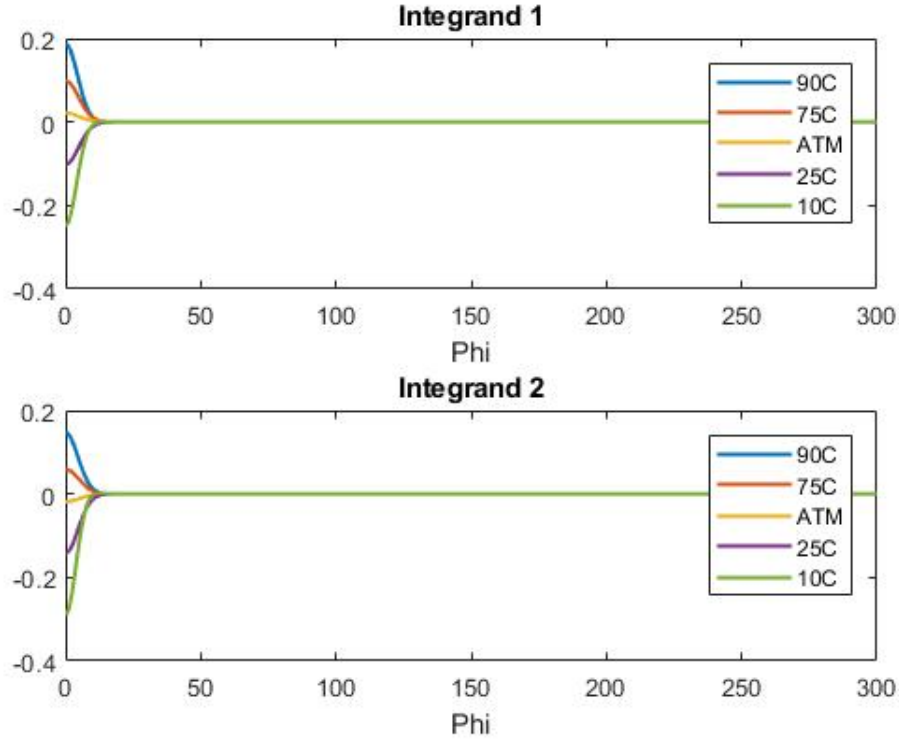


FIGURE 24. 1Y Benchmark Integral using NLLS parameters

negative value are far lower. Since this study's aim is based on numerical empirical evidence rather than a mathematical demonstration, the 4.901 results obtained from each pricing method are used to support the claims previously made in the paragraph.

## 11. CONCLUSIONS

All the pricing model were calculated at least 5 times, that implies that the study calculates . In all of them all the calculated prices of Heston(1993) and consolidating the Heston integral were positive. That is a positive result due to the amount of time the price of the stochastic model was compute.

The model that least the last were consolidating the Heston integral because it does calculate one integral. In the other hand, the numerical method that have the smallest MSE is a draw between G-Legendre and G-Laguerre because its closeness is less than  $e^{-5}$ . The remaining quadrature behaved the same way or with more error than conventional methods.

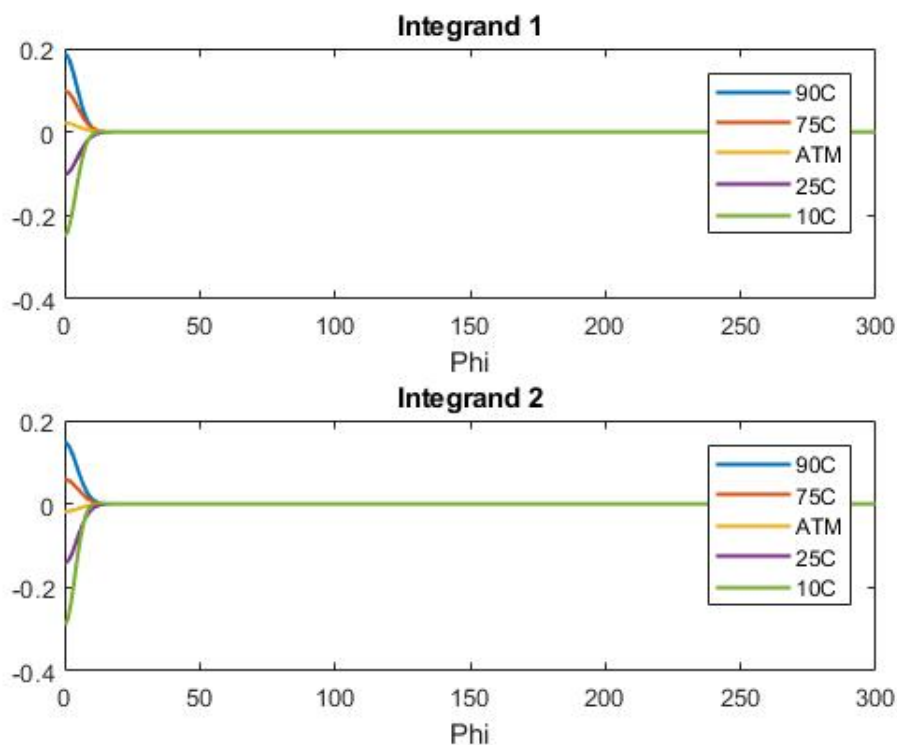


FIGURE 25. 1Y Benchmark Integral using NLLS parameters

-----Benchmark NLLS ATM Prices-----	
Maturity	Price
1W	26.302813748948437932995148003101348876953125000000000
1M	54.620268288163288161740638315677642822265625000000000
2M	73.959018061740607663523405790328979492187500000000000
3M	88.046201284274957288289442658424377441406250000000000
6M	119.731799749003812394221313297748565673828125000000000
9M	141.036473403920354030560702085494995117187500000000000
1Y	161.687953675607786863110959529876708984375000000000000
-----Benchmark MLE ATM Prices-----	
Maturity	Price
1W	24.869869286955690768081694841384887695312500000000000
1M	79.560093897677916174870915710926055908203125000000000
2M	90.601439204803227767115458846092224121093750000000000
3M	113.951246864607810493907891213893890380859375000000000
6M	160.432873063021133930305950343608856201171875000000000
9M	199.481501422470273610088042914867401123046875000000000
1Y	234.134244417504760349402204155921936035156250000000000
~~	

FIGURE 26. Benchmark ATM Price using MLE and NLLS parameters



	Mid-Point	Trapezoid al rule	Simpson's rule	Simpson's 3/8 rule	Gauss- Laguerre	Gauss- Legendre	Gauss- Lobato
<b>Heston (1993)</b>	Error 1,1	Error 1,2	Error 1,3	Error 1,4	Error 1,5	Error 1,6	Error 1,7
<b>Consolidating the integrals</b>	Error 2,1	Error 2,2	Error 2,3	Error 2,4	Error 2,5	Error 2,6	Error 2,7
<b>Carr and Madan (1997)</b>	Error 3,1	Error 3,2	Error 3,3	Error 3,4	Error 3,5	Error 3,6	Error 3,7
<b>C1 and (v) C2 fundamental transform</b>	Error 4,1	Error 4,2	Error 4,3	Error 4,4	Error 4,5	Error 4,6	Error 4,7
<b>volatility of volatility series expansion C1</b>	Error 5,1	Error 5,2	Error 5,3	Error 5,4	Error 5,5	Error 5,6	Error 5,7
<b>volatility of volatility series expansion C2</b>	Error 6,1	Error 6,2	Error 6,3	Error 6,4	Error 6,5	Error 6,6	Error 6,7

FIGURE 27. Error Matrix

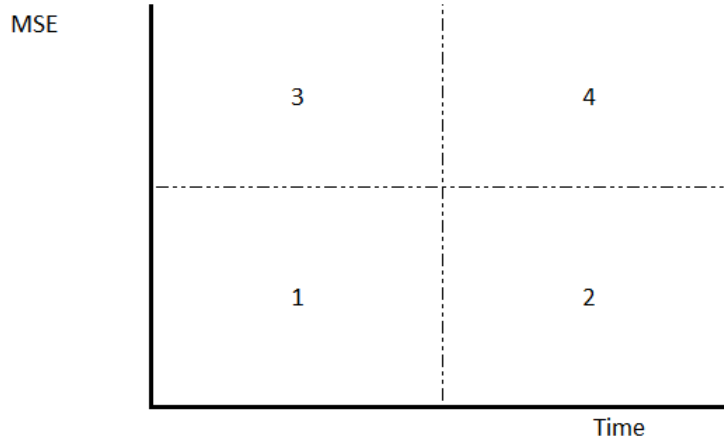


FIGURE 28. MSE Vs Time

The author recommends doing the calculations with Gauss Laguerre because it is much easier to implement and does not have to make a conversions of the limits of the integral.

In conclusion it is recommended, for the Colombian market, with USD-COP data to calculate prices with Consolidating the Integral and GL.

#### APPENDIX A. OTHER HESTON(1993) EQUIVALENT PRICING METHODS

This section present other equivalent pricing to Heston (1993)

**A.1. Carr and Madan (1999).** This pricing method incorporates a damping factor,  $\alpha$ , to price the option and use a Fast Fourier Transform (hereafter FFT) to reduce the computational time. Even though the computational time is reduced, this study does not use the FFT as a

1W	2W	3W	1M	2M	3M	1Y
Rep 1	Rep 1	Rep 1	Rep 1	Rep 1	Rep 1	Rep 1
Rep 2	Rep 2	Rep 2	Rep 2	.	.	.
Rep 3	Rep 3	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
.	.	.	.	.	.	.
Rep 98	Rep 98	Rep 99	Rep 99	Rep 100	Rep 100	Rep 100
Rep 99	Rep 99	Rep 100	Rep 100	.	.	.
Rep 100	Rep 100	.	.	.	.	.

FIGURE 29. Tiempo ATM

numerical scheme or option pricing method since the cost limitation is too high in the sense of loss of accuracy and handling. It entails a trade-off between the grid sizes, consequently, one cannot willingly choose the integration grid. This restriction is due to the constraint  $\lambda\eta = \frac{2\pi}{N}$  that entails the relationship between the integration grid and the log-strike grid. Other approaches like the Fractional FFT applied by Chourdakis (2004) [Cho04] try to solve the limitations of FFR by relaxing the restrictive constraint.

Carr and Madan (1999) uses a Fourier transform approach to option pricing. The advantage of their method is the use of only one integral that decays faster than Heston (1993), in consequence the computational time is reduced. See Lord and Kahl (2007) [LK07] for more advantages of the approach. Carr-Madan shows that the analytical solution to price a European call can be obtained once the CF is known. This method notices that discounting the present value of the payoff call,  $C(x)$ , is not integrable  $L^1$ , where  $x = \ln k$ , and therefore the following modified call price,  $c(x)$ , is proposed:

$$(A.1.1) \quad c(x) = e^{\alpha x} C(x)$$

where  $e^{\alpha x}$  is the damping factor and is applied on  $C(x)$  to make the call price integrable  $L^1$  and therefore the Fourier transform,  $\hat{c}(x)$ , for  $c(x)$  can be found.

$$(A.1.2) \quad \hat{c}(x) = \frac{e^{-r_d \tau} \varphi(v - (\alpha + 1)i)}{\alpha^2 + \alpha - v^2 + iv(2\alpha + 1)}$$

where  $\varphi$  is  $f_2(\phi)$ . To get the call price the inverse Fourier transform must be used to recover  $c(x)$  and remove the damping factor to restore the call price,  $C(x)$ .

$$(A.1.3) \quad C(x) = \frac{e^{-\alpha x}}{\pi} \int_0^\infty \text{Re} [e^{ivx} \hat{c}(v)] dv$$

The main disadvantage of this approach is having to appropriately chose the damping factor  $\alpha$  since accuracy will be lost if the parameter is distant from the real one. This study uses Lee (2004a) [L<sup>+</sup>04] to find a range of admissible values for  $\alpha$  and implements the Lord and Kahl (2007) method to find the optimal  $\alpha^*$ . To implement Carr-Madan price, one must replaced the spot price  $S$  by  $Se^{-r_f \tau}$ , see Whaley (2006) [Wha06] for justification.

This study uses the following sniped code 32 to calculate the integral

**A.2. Lewis (2000) Fundamental Transform.** This approach requires that the fundamental transform (hereafter FT),  $\hat{H}(k, v, \tau)$ , and the generalized Fourier transform of the option payoff, be available. Lewis 2000 [L<sup>+</sup>00] says: "The FT is determined by the volatility process and not by the particulars of any option contract", and defines the FT as an analytic characteristic function with all its properties. This means that the FT is modeled without a contract dependency and that the approach will only work with the Heston FT and not its CF. Through Lewis' book, some steps to follow for option pricing<sup>13</sup> are presented. Before these steps may be followed, one must obtain the generalized Fourier transform of the option payoff, which is easier to obtain than the option price itself, and the fundamental transform of the Heston model. It is worth mentioning that the fundamental transform of the Heston model is homologous to the CF.

---

<sup>13</sup>See page 39 of Lewis (2000) book

The generalized Fourier transform, unlike the regular Fourier transform, allows some arguments to be complex. The generalized Fourier transform at maturity,  $\hat{f}(k, T)$  is call the payoff transform

$$(A.2.1) \quad \hat{f}(k, T) = -\frac{K^{ik+1}}{k^2 - ik}$$

where  $k_i > 1$  for a call option. The FT of the Heston model is not straightforward, see Lewis (2000) for details of the derivation

$$(A.2.2) \quad \begin{aligned} \hat{H}(k, v, \tau) &= \exp(C_t + D_t v) \\ C_t &= \tilde{\kappa} \tilde{\theta} \left[ \frac{\tilde{\kappa} + d}{2} t - \ln \left( \frac{1 - g e^{dt}}{1 - g} \right) \right] \\ D_t &= \frac{\tilde{\kappa} + d}{2} \left( \frac{1 - e^{dt}}{1 - g e^{dt}} \right) \end{aligned}$$

where  $\tilde{\kappa} = \frac{2(\kappa + ik\rho\sigma)}{\sigma^2}$ ,  $\tilde{\theta} = \frac{\kappa\theta}{\kappa + ik\rho\sigma}$ ,  $d = \sqrt{\tilde{\kappa} + 4\tilde{c}}$ ,  $\tilde{c} = \frac{(k^2 - ik)}{\sigma^2}$ ,  $g = \frac{\tilde{\kappa} + d}{\tilde{\kappa} - d}$ . Note that the FT,  $\hat{H}(k, v, \tau)$  (A.2.2) is equal to the Heston CF (2.1.6) with the notation  $C_t$  and  $D_t$  denoting  $C_j(\tau, \phi)$  and  $D_j(\tau, \phi)$ . Once the payoff (A.2.1) and the FT (A.2.2) are known, the Lewis steps are straightforward: (i) multiply the fundamental transform (A.2.2), the payoff transform (A.2.1) and the expression  $\exp([-r_d - ik(r_d - r_f)]\tau)$ ; (ii) pass the result through the generalized inverse Fourier transform (A.2.3)

$$(A.2.3) \quad f(x, t) = \frac{1}{2\pi} \int_{ik_i - \infty}^{ik_i + \infty} e^{ikx} \hat{f}(k, t) dk$$

where  $x = \ln S_t$ . (iii) evaluate the integral over the correct strips of regularity, for which the FT,  $\hat{H}(k, v, \tau)$ , and payoff,  $\hat{f}(x, T)$  would be:

$$(A.2.4) \quad \begin{aligned} \hat{H}(k, v, \tau) &:= \frac{-\tilde{\kappa} + d}{2} < k_i < \frac{-\tilde{\kappa} - d}{2} \\ \hat{f}(x, T) &:= 1 < k_i \end{aligned}$$

Hence, the call price,  $C_1(K)$ , is:

$$(A.2.5) \quad C_1(K) = -\frac{K e^{-r_d \tau}}{\pi} \int_0^\infty \operatorname{Re} \left[ e^{ikX} \frac{1}{k^2 - ik} \hat{H}(k, v, \tau) \right] dk$$

where  $X = \ln(S/K) + (r_d - r_f)$  and  $k_i > 1$ .

Lewis (2000) establishes another way to compute the call price based on put-call parity and a covered call, for which the payoff is  $\min(S_T, K)$ . The respective covered call payoff transform is  $\hat{f}(k, T) = K^{ik+1} / (k^2 - ik)$  and the  $FT$  is (A.2.2). Upon completion of the mentioned steps<sup>14</sup> for the option pricing with the covered call, one can get:

(A.2.6)

$$C_2(K) = S_t e^{-r_f \tau} - \frac{K e^{-r_d \tau}}{\pi} \int_0^\infty \operatorname{Re} \left[ e^{ikX} \frac{1}{k^2 - ik} \hat{H}(k, v, \tau) \right] dk$$

Note that the values of  $C_1$  and  $C_2$  are the same, even though the expressions are virtually identical, the different strips make the value of the integrals distinct but operating with the remaining terms makes the prices equal.

### A.3. Lewis (2000) Volatility of Volatility Series Expansion.

Pricing with the Heston model by the Vol. of Vol. expansion is computationally faster because the approach does not solve an integral, like the name says, the price is obtained by a series of sums. As a matter of fact, the expansion derived its name from the powers of the volatility of volatility parameter,  $\sigma$ , from which the series are expressed in. There are two expansions, one for the price and one for the implied volatility, known as Series I and II respectively.

Series I is based on the  $BSM$  price

$$(A.3.1) \quad C_{BS}(S_0, \bar{v} T) = S_0 e^{-r_f T} \phi(d_1) - K e^{-r_d T} \phi(d_2)$$

where  $d_1 = (\log(S_0/K) + (r_d - r_f + \bar{v}^2/2)T) / (\sqrt{\bar{v}T})$  and  $d_2 = d_1 - \sqrt{\bar{v}T}$ . The expression,  $\bar{v}$ , is the expected average variance over the lifetime of the option,  $(0, T)$ . It is defined as:

(A.3.2)

$$\begin{aligned} \bar{v} &= E \left[ \frac{1}{T} \int_0^T v_t dt \middle| v_0 \right] = \frac{1}{T} \int_0^T E[v_t | v_0] dt \\ &= \frac{1}{T} \int_0^T v_t [\theta + (v_0 - \theta) e^{-\kappa T}] dt = (v_0 - \theta) \left( \frac{1 - e^{-\kappa T}}{\kappa T} \right) + \theta \end{aligned}$$

Series I also uses the  $BSM$  Vega evaluated at the expected average variance,  $\bar{v}$ . Note the importance of,  $\bar{v}$ , in the derivation of this approach.

---

<sup>14</sup>The strip for step 3 is  $k_i = 1/2$

$$(A.3.3) \quad C_v(S_0, \bar{v}, T) = \left. \frac{\partial C_{BSM}}{\partial v} \right|_{v=\bar{v}} = \sqrt{\frac{T}{8\pi\bar{v}}} S_0 e^{-r_f T} \exp\left(-\frac{1}{2}d_1^2\right)$$

As mentioned, the first series calculates the Heston price directly, meanwhile the second one provides the implied variance that serves as input in the *BSM* pricing formula to compute the Heston price. Series I is (A.3.4a) while Series II is (A.3.4b).

(A.3.4a)

$$\begin{aligned} C_I(S_0, v_0, T) \approx & C_{BS}(S_0, \bar{v}_0, T) + \sigma \frac{J_1}{T} R^{1,1} C_v(S_0, v_0, T) + \\ & \sigma C_v(S_0, v_0, T) \left[ \frac{J_3 R^{2,0}}{T^2} + \frac{J_4 R^{1,2}}{T} + \frac{(J_1)^2 R^{2,2}}{2T^2} \right] \end{aligned}$$

(A.3.4b)

$$\begin{aligned} C_{II}(S_0, v_0, T) &= C_{BS}(S_0, v_{imp}, T) \\ v_{imp} &\approx \bar{v} + \frac{J_1 R^{1,1}}{T} + \\ &\quad \sigma^2 \left[ \frac{J_3 R^{2,0}}{T^2} + \frac{J_4 R^{1,2}}{T} + \frac{(J_1)^2}{2T^2} (R^{2,2} - R^{2,0} (R^{1,1})^2) \right] \end{aligned}$$

Both series depend on the expected average variance,  $\bar{v}$ , as well as in the  $J_{1,3,4}$  and  $R$  equations. Lewis (2000) explains that  $J_2$  equation vanishes since the drift is linear. Each  $J_s$  equation presented here is a solution of a respective integral with  $\phi(1/2)$ . The interested reader may check the integrals in Lewis 2000 for further details.

(A.3.5)

$$\begin{aligned} J_1(v_0, T) &= \frac{\rho}{\kappa} \left[ \theta T + (1 - e^{-kT}) \left( \frac{v_0}{k} - \frac{2\theta}{k} \right) - e^{-kT} (v_0 - \theta) T \right] \\ J_3(v_0, T) &= \frac{\theta}{2\kappa^2} \left[ T + \frac{1}{2\kappa} (1 - e^{-2\kappa T}) - \frac{2}{\kappa} (1 - e^{-kT}) \right] \\ &\quad + \frac{(v_0 - \theta)}{2\kappa^2} \left[ \frac{1}{k} (1 - e^{-2kT}) - 2T e^{-kT} \right] \\ J_4(v_0, T) &= \frac{\rho^2 \theta}{\kappa^3} \left[ T (1 + e^{-\kappa T}) - \frac{2}{\kappa} (1 - e^{-\kappa T}) \right] \\ &\quad - \frac{\rho^2}{2\kappa^2} T^2 e^{-\kappa T} (v_0 - \theta) + \frac{\rho^2 (v_0 - \theta)}{\kappa^3} \left[ \frac{1}{\kappa} (1 - e^{-\kappa T}) - T e^{-\kappa T} \right] \end{aligned}$$

Finally, the  $R$  equations are presented. They are  $BSM$  ratios used to compute the Heston price.

$$\begin{aligned}
 R^{1,1} &= \left[ \frac{1}{2} - W \right], \quad R^{1,2} = \left[ W^2 - W - \frac{4 - Z}{4Z} \right] \\
 (A.3.6) \quad R^{2,0} &= T \left[ \frac{W^2}{2} - \frac{1}{2Z} - \frac{1}{8} \right] \\
 R^{2,2} &= T \left[ \frac{W^4}{2} - \frac{W^3}{2} - \frac{3X^2}{Z^3} + \frac{X(12 + Z)}{8Z^2} + \frac{48 - Z^2}{32Z^2} \right]
 \end{aligned}$$

where  $W = X/Z$ ,  $X = \log(S_0/K) + (r_d - r_f)T$ ,  $Z = \bar{v}T$ .

## BIBLIOGRAPHY

- [Ale08] C. Alexander, *Market risk analysis, pricing, hedging and trading financial instruments*, Market Risk Analysis, Wiley, 2008.
- [AMS] H Albrecher, P Mayer, and W Tistaert Schoutens, *The little heston trap*, Wilmott Magazine, January issue, 83–92.
- [AS64] Milton Abramowitz and Irene A Stegun, *Handbook of mathematical functions: with formulas, graphs, and mathematical tables*, vol. 55, Courier Corporation, 1964.
- [AW09] Amir F Atiya and Steve Wall, *An analytic approximation of the likelihood function for the heston model volatility estimation problem*, Quantitative Finance **9** (2009), no. 3, 289–296.
- [Bac05] K. Back, *A course in derivative securities: Introduction to theory and computation*, Springer Finance, Springer Berlin Heidelberg, 2005.
- [Bat96] David S Bates, *Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options*, Review of financial studies **9** (1996), no. 1, 69–107.
- [BF10] R.L. Burden and J.D. Faires, *Numerical analysis*, Cengage Learning, 2010.
- [BGZ11] Tim Bollerslev, Michael Gibson, and Hao Zhou, *Dynamic estimation of volatility risk premia and investor risk aversion from option-implied and realized volatilities*, Journal of econometrics **160** (2011), no. 1, 235–245.
- [BIB<sup>+</sup>15] E. Berlinger, F. Illés, M. Badics, Á. Banai, G. Daróczy, B. Dömötör, G. Gabler, D. Havran, P. Juhász, I. Margitai, et al., *Mastering r for quantitative finance*, Community experience distilled, Packt Publishing, 2015.
- [Bjö09] T. Björk, *Arbitrage theory in continuous time*, Oxford Finance Series, OUP Oxford, 2009.
- [BM00] Gurdip Bakshi and Dilip Madan, *Spanning and derivative-security valuation*, Journal of financial economics **55** (2000), no. 2, 205–238.
- [BR10] Claus Christian Beier and Christoph Renner, *Foreign exchange options: Delta-and at-the-money conventions*, Encyclopedia of Quantitative Finance (2010).

- [BRSD10] Frédéric Bossens, Grégory Rayée, Nikos S Skantzios, and Griselda Deelstra, *Vanna-volga methods applied to fx derivatives: from theory to market practice*, International journal of theoretical and applied finance **13** (2010), no. 08, 1293–1324.
- [Cas10] A. Castagna, *Fx options and smile risk*, The Wiley Finance Series, Wiley, 2010.
- [CHJ09] Peter Christoffersen, Steven Heston, and Kris Jacobs, *The shape and term structure of the index option smirk: Why multifactor stochastic volatility models work so well*, Management Science **55** (2009), no. 12, 1914–1932.
- [Cho04] Kyriakos Chourdakis, *Option pricing using the fractional fft*, Journal of Computational Finance **8** (2004), no. 2, 1–18.
- [Cla11] I.J. Clark, *Foreign exchange option pricing: A practitioner's guide*, The Wiley Finance Series, Wiley, 2011.
- [Coh15] G. Cohen, *The bible of options strategies: The definitive guide for practical trading strategies*, Pearson Education, 2015.
- [Con10] R. Cont, *Encyclopedia of quantitative finance, 4 volume set*, Wiley, 2010.
- [CS89] Marc Chesney and Louis Scott, *Pricing european currency options: A comparison of the modified black-scholes model and a random variance model*, Journal of Financial and Quantitative Analysis **24** (1989), 267–284.
- [FB73] Myron Scholes Fischer Black, *The pricing of options and corporate liabilities*, Journal of Political Economy **81** (1973), no. 3, 637–654.
- [Gla04] P. Glasserman, *Monte carlo methods in financial engineering*, Applications of mathematics : stochastic modelling and applied probability, Springer, 2004.
- [GP51] J Gil-Pelaez, *Note on the inversion theorem*, Biometrika **38** (1951), no. 3-4, 481–482.
- [GT11] J. Gatheral and N.N. Taleb, *The volatility surface: A practitioner's guide*, Wiley Finance, Wiley, 2011.
- [Haf04] R. Hafner, *Stochastic implied volatility: A factor-based model*, Lecture Notes in Economics and Mathematical Systems, Springer Berlin Heidelberg, 2004.
- [Hes93] Steven L Heston, *A closed-form solution for options with stochastic volatility with applications to bond and currency options*, Review of financial studies **6** (1993), no. 2, 327–343.
- [HKLW02] Patrick S Hagan, Deep Kumar, Andrew S Lesniewski, and Diana E Woodward, *Managing smile risk*, The Best of Wilmott (2002), 249.
- [HN13] A. Hirsu and S.N. Neftci, *An introduction to the mathematics of financial derivatives*, Elsevier Science, 2013.
- [Hul12] J. Hull, *Options, futures, and other derivatives*, Options, Futures, and Other Derivatives, Prentice Hall, 2012.
- [HW87] John Hull and Alan White, *The pricing of options on assets with stochastic volatilities*, The journal of finance **42** (1987), no. 2, 281–300.
- [Iac11] S.M. Iacus, *Option pricing and estimation of financial models with r*, Wiley, 2011.



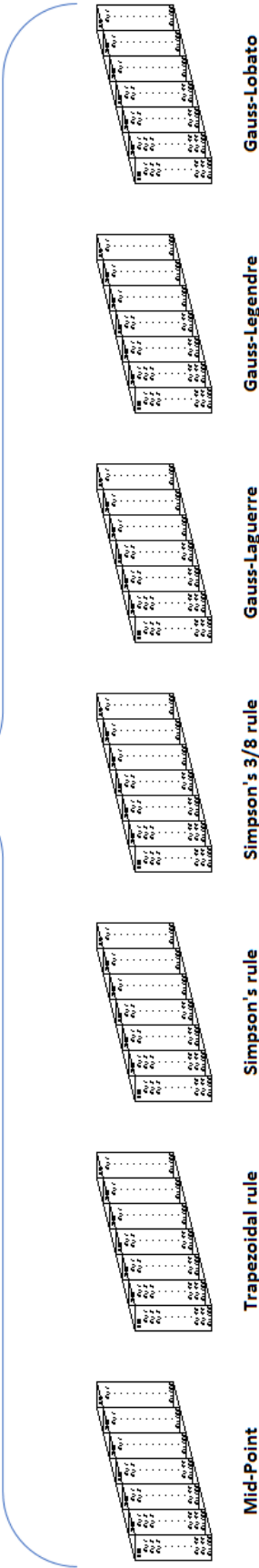
- [JYC09] M. Jeanblanc, M. Yor, and M. Chesney, *Mathematical methods for financial markets*, Springer Finance, Springer London, 2009.
- [Kil06] Fiodar Kilin, *Accelerating the calibration of stochastic volatility models*.
- [KJ05] Christian Kahl and Peter Jäckel, *Not-so-complex logarithms in the heston model*, Wilmott magazine **19** (2005), no. 9, 94–103.
- [L<sup>+</sup>00] Alan L Lewis et al., *Option valuation under stochastic volatility*, Option Valuation under Stochastic Volatility (2000).
- [L<sup>+</sup>04] Roger W Lee et al., *Option pricing by transform methods: extensions, unification and error control*, Journal of Computational Finance **7** (2004), no. 3, 51–86.
- [LK07] Roger Lord and Christian Kahl, *Optimal fourier inversion in semi-analytical option pricing*.
- [Lue98] David G. Luenberger, *Investment science*, Oxford University Press, 1998.
- [MCC98] Dilip B Madan, Peter P Carr, and Eric C Chang, *The variance gamma process and option pricing*, European finance review **2** (1998), no. 1, 79–105.
- [Nat94] S. Natenberg, *Option volatility & pricing: Advanced trading strategies and techniques: Advanced trading strategies and techniques*, McGraw-Hill Education, 1994.
- [Rei10] Dimitri Reiswich, *The foreign exchange volatility surface*, Ph.D. thesis, Frankfurt School of Finance & Management, 2010.
- [RH13] F.D. Rouah and S.L. Heston, *The heston model and its extensions in matlab and c#*, Wiley Finance, Wiley, 2013.
- [RV12] F.D. Rouah and G. Vainberg, *Option pricing models and volatility using excel-vba*, Wiley Finance, Wiley, 2012.
- [Sco87] Louis O. Scott, *Option pricing when the variance changes randomly: Theory, estimation, and an application*, Journal of Financial and Quantitative Analysis **22** (1987), 419–438.
- [Shr04] S.E. Shreve, *Stochastic calculus for finance ii: Continuous-time models*, Springer Finance Textbooks, no. v. 11, Springer, 2004.
- [SS91] Elias M Stein and Jeremy C Stein, *Stock price distributions with stochastic volatility: an analytic approach*, Review of financial Studies **4** (1991), no. 4, 727–752.
- [Sto13] Santiago Stozitzky, *General bounds for arithmetic asian option prices: Colombian fx option market application*, Master’s thesis, The University of Edinburgh, 2013.
- [SZ99] Rainer Schöbel and Jianwei Zhu, *Stochastic volatility with an ornstein-uhlenbeck process: an extension*, European Finance Review **3** (1999), no. 1, 23–46.
- [Web11] N. Webber, *Implementing models of financial derivatives: Object oriented applications with vba*, The Wiley Finance Series, Wiley, 2011.
- [Wha06] Robert E Whaley, *Derivatives: Markets, Valuation, and Risk Management* **345** (2006).
- [Wig87] James B Wiggins, *Option values under stochastic volatility: Theory and empirical estimates*, Journal of financial economics **19** (1987), no. 2, 351–372.

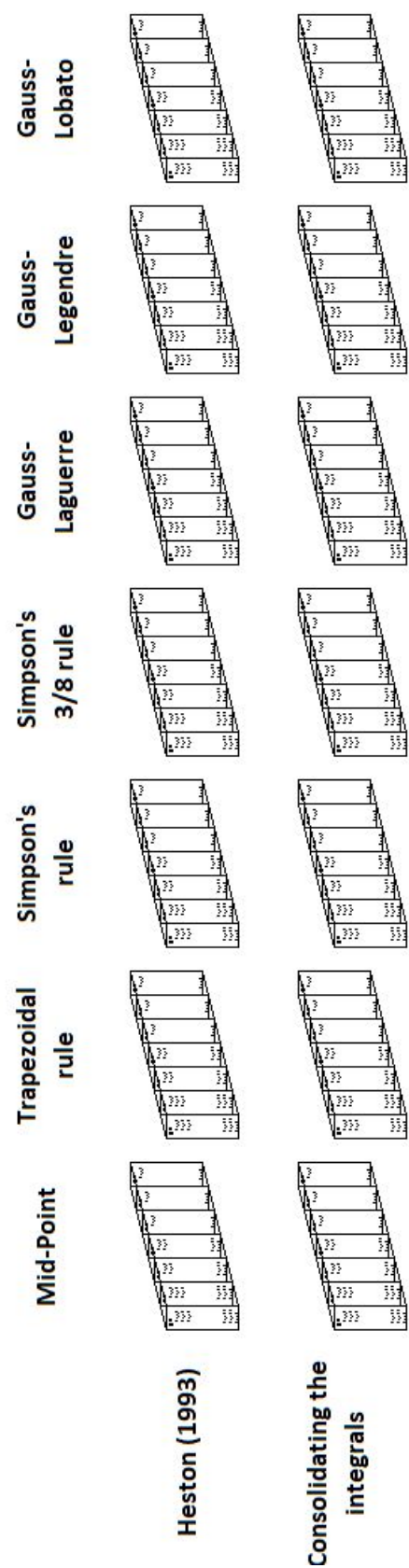
- [Wil06] P. Wilmott, *Paul wilmott on quantitative finance, 3 volume set*, Paul Wilmott on Quantitative Finance, Wiley, 2006.
- [Wil13] ———, *Paul wilmott introduces quantitative finance*, The Wiley Finance Series, Wiley, 2013.
- [Wys07] U. Wystup, *Fx options and structured products*, The Wiley Finance Series, Wiley, 2007.
- [Zhu09] J. Zhu, *Applications of fourier transform to smile modeling: Theory and implementation*, Springer Finance, Springer Berlin Heidelberg, 2009.

(Javier Lázaró) CR 51A No127-49, BOGOTÁ, COLOMBIA.

*E-mail address:* javierlazaró1@hotmail.com

Pricing Method





```

function y = Consoli_Integrand(phi,kappa,theta,lambda,rho,sigma,tau,K,S,rd,rf,v0)
% Returns the integrand for the risk neutral probabilities P1 and P2

% Log of the stock price
x = log(S);

% Parameter "a" is the same for P1 and P2.
a = kappa*theta;

% First characteristic function f1
u1 = 0.5;
b1 = kappa + lambda - rho*sigma;
d1 = sqrt((rho*sigma*i*phi - b1)^2 - sigma^2*(2*u1*i*phi - phi^2));
g1 = (b1 - rho*sigma*i*phi + d1) / (b1 - rho*sigma*i*phi - d1);

% "Little Heston Trap" formulation
c1 = 1/g1;
D1 = (b1 - rho*sigma*i*phi - d1)/sigma^2*((1-exp(-d1*tau))/(1-c1*exp(-d1*tau)));
G1 = (1 - c1*exp(-d1*tau))/(1-c1);
C1 = (rd-rf)*i*phi*tau + a/sigma^2*((b1 - rho*sigma*i*phi - d1)*tau - 2*log(G1));

f1 = exp(C1 + D1*v0 + i*phi*x);

% Second characteristic function f2
u2 = -0.5;
b2 = kappa + lambda;
d2 = sqrt((rho*sigma*i*phi - b2)^2 - sigma^2*(2*u2*i*phi - phi^2));
g2 = (b2 - rho*sigma*i*phi + d2) / (b2 - rho*sigma*i*phi - d2);

% "Little Heston Trap" formulation
c2 = 1/g2;
D2 = (b2 - rho*sigma*i*phi - d2)/sigma^2*((1-exp(-d2*tau))/(1-c2*exp(-d2*tau)));
G2 = (1 - c2*exp(-d2*tau))/(1-c2);
C2 = (rd-rf)*i*phi*tau + a/sigma^2*((b2 - rho*sigma*i*phi - d2)*tau - 2*log(G2));

f2 = exp(C2 + D2*v0 + i*phi*x);

% Return the real part of the integrand.
y = real(exp(-i*phi*log(K))/i/phi*(S*exp(-rf*tau)*f1 - K*exp(-rd*tau)*f2));

```

FIGURE 32. Error Matrix